TAMPERE UNIVERSITY OF TECHNOLOGY

Antti Ainasoja

VIDEO SUMMARIZATION WITH KEY FRAMES

Master's Thesis

Examiners: Professor Joni-Kristian Kämäräinen

Examiners and topic approved in the Computing and Electrical Engineering Faculty Council meeting on 6th April 2016

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY
Master's Degree Programme in Signal Processing and Communications Engineering
ANTTI AINASOJA: **Video summarization with key frames**
Master of Science Thesis, 42 pages
April 2016
Major: Multimedia
Examiner: Professor Joni-Kristian Kämäräinen
Keywords: video summary, key frame detection, skimming, optical flow, motion analysis

Video summarization is an important tool for managing and browsing video content. The increasing amount of consumer level video recording devices combined with the availability of cheap high bandwidth internet connections have enabled ordinary people to become video content producers and publishers. This has resulted in massive increase in online video content. Tools are needed for efficiently finding relevant content devoid traditional viewing.

Video summaries provide a condensed view of the actual video. They are most commonly presented as static still images in the form of storyboards or dynamic video skims, which are shorter versions of the actual videos. Although methods for creating summaries with the assistance of computers have been long studied, practical implementations of the summarization methods are only a few.

In this thesis, a semi-supervised workflow and a tool set for creating summaries is implemented. At first, the implemented tool creates a static storyboard summary of an input video automatically. Users are able to use the storyboard summaries to select the most important content and the selected content is then used to create a video skim.

Major part of the thesis work consists of evaluating and finding the best methods to detect single key frames that would best depict the contents of a video. The evaluation process is focused mainly on motion analysis based optical flow histograms.

In the experimental part, the performance of the implemented workflow is compared to state of the art automatic video summarization method. Based on the experiment results, even a rather simple method can produce good results and keeping the human in the loop for key frame selection is beneficial for generating meaningful video summaries.

TIIVISTELMÄ

TAMPEREEN TEKNILLINEN YLIOPISTO
Tietoliikenteen ja signaalinkäsittelyn koulutusohjelma
ANTTI AINASOJA: **Videoyhteenvedon koostaminen avainruutujen avulla**
Diplomityö, 42 sivua
Huhtikuu 2016
Pääaine: Multimedia
Tarkastaja: Professori Joni-Kristian Kämäräinen
Avainsanat: videoyhteenveto, avainruudun etsiminen, lyhenyäminen, optinen vuo, liikeanalyysi

Lyhyet videotiivistelmät tarjoavat kompaktin näkymän pidemmästä videosta. Tiivistelmät esitetään yleisimmin staattisina kuvasarjoina tai dynaamisina videolyhennelminä. Vaikka tietokoneavusteisia menetelmiä videotiivistelmien tuottamiseksi on tutkittu paljon, niin käytännön toteutuksia löytyy vain vähän.

Tässä työssä toteutetaan puoliautomaattinen työkalu videotiivistelmien koostamiseksi. Toteutettu sovellus tuottaa aluksi automaattisesti staattisen kuvasarjan alkuperäisestä videosta. Käyttäjät voivat tämän jälkeen valita kuvasarjasta avainkuvat, jota he pitävät tärkeänä. Lopuksi valitun sisällön perusteella luodaan dynaaminen videolyhennelmä.

Suurin osa tehdystä työstä on eri vaihtoehtojen arvioimista parhaimpien menetelmien löytämiseksi. Menetelmien avulla saadaan automaattisesti videon sisältöä hyvin kuvaavat avainruudut. Menetelmien arvioinnissa keskitytään erityisesti optiseen vuohon perustuvaan liikeanalyysiin.

Toteutettua menetelmää verrataan testausosiossa videoyhteenvedon automaattisesti koostavaan huippumenetelmään. Vertailutestien perusteella yksinkertaisellakin menetelmällä voidaan saavuttaa hyviä tuloksia ja ihmisen pitäminen mukana avainruutujen valinnassa on hyödyllistä sisällöllisesti merkityksellisten videolyhennelmien tuottamisessa.

PREFACE

I would like to thank Professor Joni Kämäräinen for giving me the topic and the opportunity to write this thesis, and for his kind supervision during the project.

"I love deadlines. I love the whooshing sound they make as they fly by." - Douglas Adams

Tampere, April 19th, 2016

Antti Ainasoja

CONTENTS

ABBREVIATIONS AND SYMBOLS

D	intersection
F	pairwise f-measure
h	frame height
H	Optical flow histogram
H_{frame}	Optical flow histogram for a frame
H_{avg}	average histogram
H_{mean}	mean histogram
H_{median}	median histogram
H_{norm}	normalized histogram
p	precision
r	recall
t_{max}	maximum duration
t_{scene}	duration of scene
t_{target}	target duration
t_{total}	total duration
v	magnitude of motion vector
V	total magnitude of motion vectors in a frame
V_{max}	maximum of total magnitudes
V_{min}	minimum of total magnitudes
w	frame width
θ	direction of motion vector
API	Application Programming Interface
BoW	bag-of-words
CSS	Cascading Style Sheets
fps	frames per second
HTML	Hyper Text Markup Language
NVENC	Nvidia video encoder
OpenCV	Open Source Computer Vision Library
SIFT	Scale-Invariant Feature Transform

1 INTRODUCTION

The amount of consumer devices capable of video recording has been increasing during the past decades. Practically every mobile phone duplicates as a video camera and the use of small egocentric action-cameras is gaining popularity [1]. With recent development in home automation even common household appliances, such as ovens, are being equipped with cameras [2].

The myriad devices capable of digital video recording combined with relatively cheap high bandwidth internet connections available everywhere have enabled ordinary people to start producing and publishing their own videos easily. This has resulted in shorter production times and the amount of user generated online video content is growing at an enormous pace [3]. The sheer amount of content makes digesting all video impossible and even finding the interesting content is becoming increasingly difficult. Tools are needed to help make use of all the video material. [4, 5]

One tool to help find and consume video content is video summarization. The idea of video summarization is to cover the important bits of the original footage, or to describe what the video is about, in brief and concise form. The resulting summaries may be used for evaluation, advertisement, content retrieval, navigation or for data reduction. Their content and form vary depending on the application [5]. If, for example, summary is used to arouse users interest and provide information about some aspects of the video, as is the case in motion picture trailers, it will look very different than it would when used to highlight key points of a lecture. The trailer needs to contain information about the motion picture without giving away plot twists and important events. The lecture highlights on the contrary needs to show specifically the important events and would look very different. Video skims and key frames are examples of common forms of video summaries [6].

Key frames are still frames extracted from the input video. Each extracted frame depicts a portion of the original video. Still frames contain no information about motion nor audio, nor do they need to be viewed sequentially in strict temporal order. This makes them efficient for giving an overall view of the video content or for navigation. Key frames can be easily displayed as storyboards such as Figure 1, slide shows or even in a form of comics or graphic novels [7, 8, 9].

Skims are short clips extracted from longer videos. Each shorter clip contains the most essential part of the portion of the video they are depicting. Skims are usually viewed sequentially. Due to presence of motion and audio they may give better understanding of

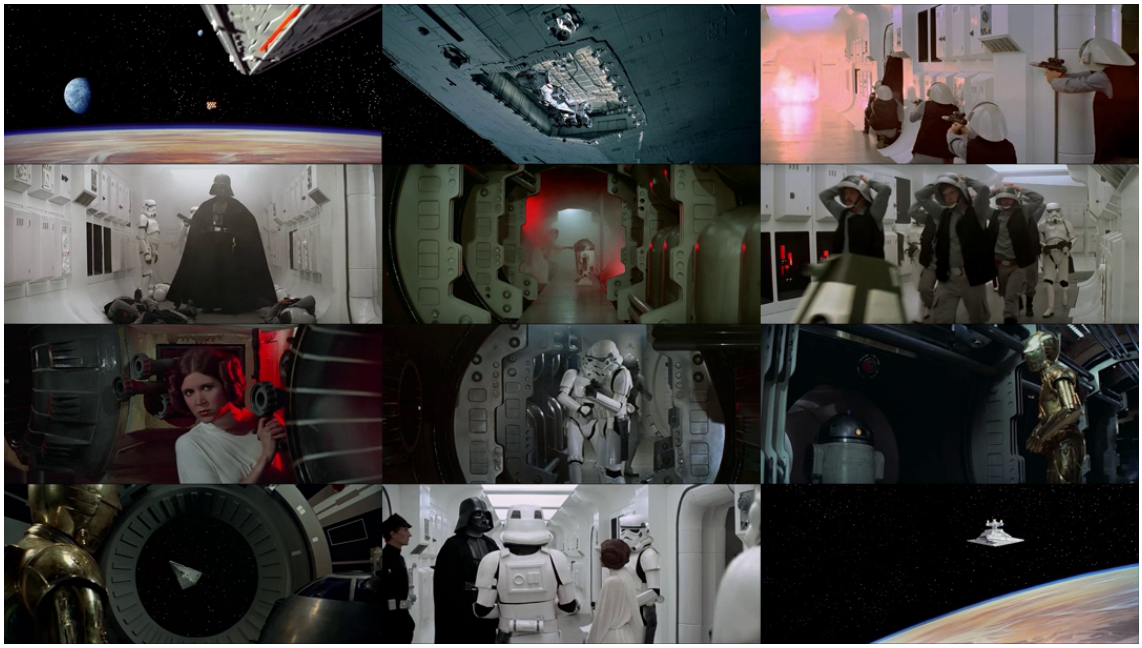


Figure 1. A static storyboard summary of the opening scene of Star Wars[1]

what is happening in the video, and they may be more fun to watch. They are however more time consuming to view and browse through than key frames. Skims and key frames can also be used together and each one can be created from the other.

In the experimental part automatic methods for detecting key frames using optical flow are investigated. Video summaries are created based on the detected key frames and the resulting summaries are compared to summaries created manually by humans. The best automatic methods are then used to create a semi-automatic approach. In semi-automatic approach human users are given a selection of key frames to choose ones they like. The summaries based on the semi-automatically selected key frames are again compared to ones created manually and by other automatic methods.

1.1 Structure of thesis

Chapter 1 gives a short overview of why research and development related to video summarization is needed. Chapter 2 provides some background on methods that are used in this work as well as what have been used by others. In Chapter 3 details on how the methods were implemented and which tools were used are described. In Chapter 4 the summarization workflow performance is evaluated. Chapter 5 includes discussion on the tools used for implementation and evaluation of the workflow. Some future development

[1]Star Wars, directed by George Lucas (1977; Lucasfilm Ltd.).

ideas are also given. In Chapter 6 a short conclusion of the work done is given.

1.2 Goals and restrictions

Although video summarization has been well studied, the complexity of methods has limited the practical implementations [10]. This thesis introduces a full semi-supervised workflow and a tool set for creating short summaries of longer videos easily. The tool can also be used as a semi-automated video editor. The goal of the work is to create the tool set in such a way that it is easy to use and that it produces good results with minimal effort.

2 THEORETICAL BACKGROUND

2.1 Related work

Automated video summarization and closely related key frame detection have long been a subjects of research. Video summarization is considered an important tool for managing and evaluating video content, but despite the work done technologies are not yet capable of producing great results.

One of the most well known works on key frame detection using motion analysis and optical flow is the algorithm by Wolf [11] from 1996. Wolf's key frame detection algorithm detects key frames within a scene by using motion analysis to find local motion minimas. The summarization method was made primarily for professionally created movies, where lack of motion is used to emphasize important moments.

Some of the more recent works on using optical flow histograms for video analysis include the approaches of Wang and Snoussi [12] and Colque et al. [13] to detect abnormal events on videos. These approaches use optical flow orientation histograms obtained from training videos to define what is considered normal behaviour in the videos. The optical flow histogram features from analysed videos are then compared to them to detect anomalies. The approaches were created primarily to detect abnormal behaviour within crowds using surveillance cameras.

The approach for key frame detection developed in this thesis is similar to the ones described above. There is however no need for training the system beforehand, as the video being analysed is used for that purpose on the fly. In addition to abnormal events the approach of this thesis attempts to find also events that are considered the most normal within the video, whether or not the video is of professional quality.

The recent works on video summarization include an approach by Gygli et al. [14], which attempts to create informative and visually pleasing summaries. The approach divides the input video into short "superframes". This is done first by dividing the video into fixed length shots and then using an iterative process to move the cut points between the parts so that the motion in the frames around them would resemble each other as closely as possible. The idea is that when a cut is made, the motion stays similar and video is less irritating to watch than when having sudden changes in movement. To determine which "superframes" are included in the summary, Gygli et al. use a combination of

various methods including low level analysis such as color information, and more evolved methods like detecting people and known landmarks.

Another summarization approach by Ejaz et al. [15] uses color gradient information to detect key frames and provide a story board summaries. This approach uses gradient information of downscaled frames to determine whether the content of the individual frame is interesting or not. This is combined with analysis on changes in the pixel intensities between frames to approximate motion information. The approach aims to be less computationally demanding than the visual attention based methods which are traditionally based on optical flow.

In comparison to the above mentioned summarization approaches, the one in this thesis is not completely automatic. It is assumed that even the modern computers today lack the intelligence to reliably determine what is important to the humans. Rather than using complex methods for detecting landmarks and people, frames with possible elements of interest are detected and the final evaluation of their importance is left to human users. The results of the summarization method in this work is compared to those by Gygli et.al and by those produced by a method based on the approach by Ejaz et al. in order to determine how well it works.

2.2 Video structure

Videos are structured spatially and temporarily. Spatial structure deals with the composition of the scene; where the objects filmed are located on the screen. For instance, a "rule of thirds" or phi grid, or centering, which are illustrated in Figure 2, are commonly used to align elements in the image.

The rule of thirds states that larger elements should occupy one third or two thirds of the vertical and/or horizontal image space and the objects of importance should be located one third of image borders both horizontally and vertically. [16]

Phi grid, also known as golden ratio, divides the space into a grid that has lines approximately 61,8 % of the width or height from the image borders. Sometimes the grid is presented in the form of Fibonacci-spiral. The elements are placed similarly to the rule of thirds or along the line of the spiral.

Although there is very little basis for using these exact rules [17, 18], they are still widely

Figure 2. Spatial composition of opening scene of Barry Lyndon[2] with fibonacci-spiral (yellow) framing the scene and skyline and actors positioned on phi-grid (green) and rule of thirds -grid (red).

used as guides to create balanced views and to draw viewers attention to the objects of importance. When recording videos the grids also help keeping the objects within the frame, especially when camera or the objects are moving. [19]

In temporal domain, videos are structured into scenes, shots, and frames. A frame is a single still image taken at a point in time. A shot is a set of consecutive frames that have been continuously recorded and a scene consists of one ore more related shots. A visualization of temporal video structure is presented in Figure 3. [20]

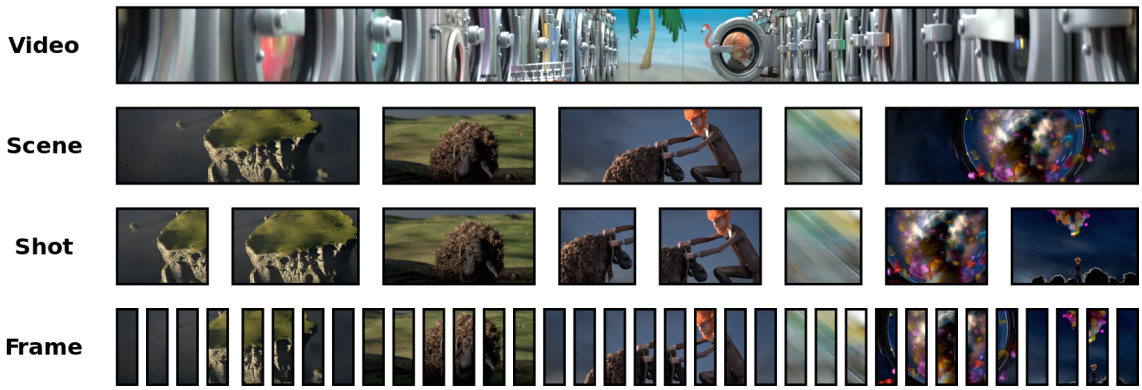


Figure 3. Temporal video structure.[3]

The video structure is not, however, always a rock solid conception. In an edited video the transitions between consecutive shots can be done by a hard cut or gradually by various transitional effects. In a hard cut the boundary between shots is clear; frame before

[2]Barry Lyndon, directed by Stanley Kubrick (1975; Warner Bros.).
[3]Screenshots from Cosmos Laundromat, directed by Mathieu Auvray (2015; Blender Foundation).

cut is part of the first shot and the next frame is part of the next shot. With the use of transitional effects this position becomes unclear. If, for example, a transition is done by crossfade, gradually dissolving parts of the shots together, the frames contain the images from both shots superimposed. It may be unclear which shot the superimposed frames belong to. The amount of different artistic transitional effects is only limited by the artists' imagination. Some examples of video transitions are presented in Figure 4.

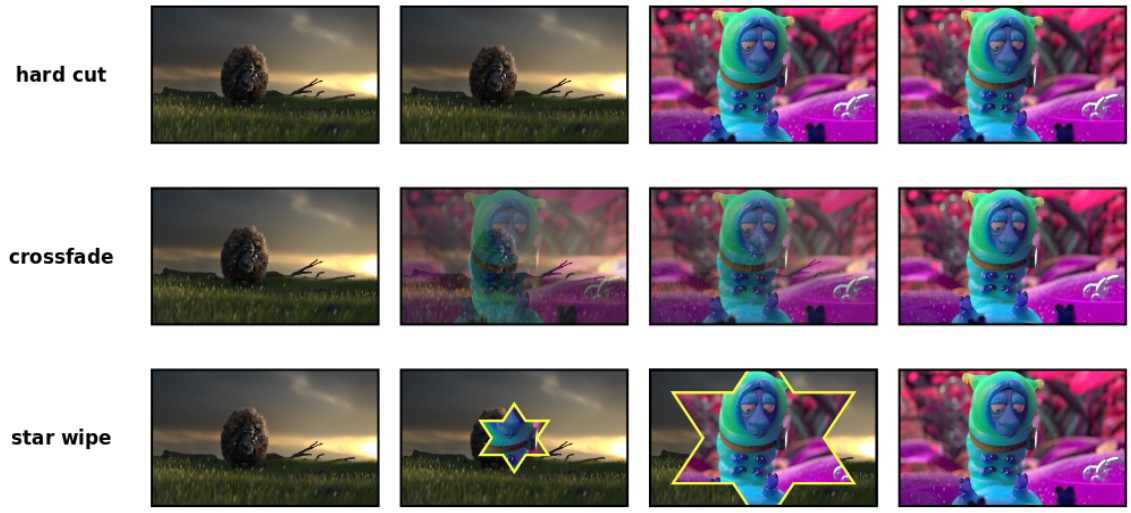


Figure 4. A visualisation of transition between shots using hard cut, cross fade and star wipe.[4]

Raw footage from consumer devices is seldom as well structured as the professionally edited works of art. The shots are in temporal order rather than merged into scenes that would convey a story, the spatial composition may be off due to lack of skill and footage may contain shots that are unusable due to camera shakes, or because camera was left recording unintentionally. [21]

2.3 Video summarization

There are a multitude of techniques developed for creating video summaries. These techniques can be divided into categories based on the summarization process as Ajmal et al. [10] proposes.

The six main categories are feature based, clustering based, event based, shot based, trajectory analysis and mosaic. In feature based techniques the summaries are generated by

[4]Screenshots from Cosmos Laundromat, directed by Mathieu Auvray (2015; Blender Foundation).

detecting and tracking features such as color, motion, objects, audio and speech. Clustering based algorithms group video frames based on similar characteristics. They are efficient for representing the contents of the video, but unusable for navigation or browsing. Event based summaries attempt to detect events happening in videos. These work well on videos that have static background, but if background is moving it could be falsely identified as an event. Shot based techniques create segmented videos by shot boundary detection and are commonly used with moving camera footage. Trajectory analysis provide information on moving objects against static backgrounds. It is used in surveillance cameras but fails with moving cameras. Mosaic summaries are panoramic images created from multiple consecutive frames. They work on videos having static backgrounds, but lose information about moving objects. [10]

2.4 Overall workflow

An overview of summarization techniques show that the use of motion analysis is a valid option for use in most video content [10]. In this work a combination of shot and motion based approaches is used to create a summarization workflow that can be use for two main purposes; To edit footage by cutting out material that is irrelevant to the user, and to create short overviews of overall contents of longer videos.

The summarization process is illustrated in Figure 5. The input video is split into scenes by scene boundary detector. Key frames are detected from each scene. Key frames are selected to decide what will be included in the summary. Selected key frames are expanded to create shots around them and final summary is created by combining the shots into one video skim.

When summary is created semi-automatically, user is presented with a set of key frames for each scene. User can select key frames that seem relevant. The skim is created by taking the footage around each selected frame according to a given target summary length. If target length is omitted, the scenes where key frames are selected are used entirely.

When summary is created automatically, it is always done according to a given target summary length. Every scene from input video is included in the resulting summary. The skimming is done by picking content around the detected key frames, resulting a shorter view of the original content.

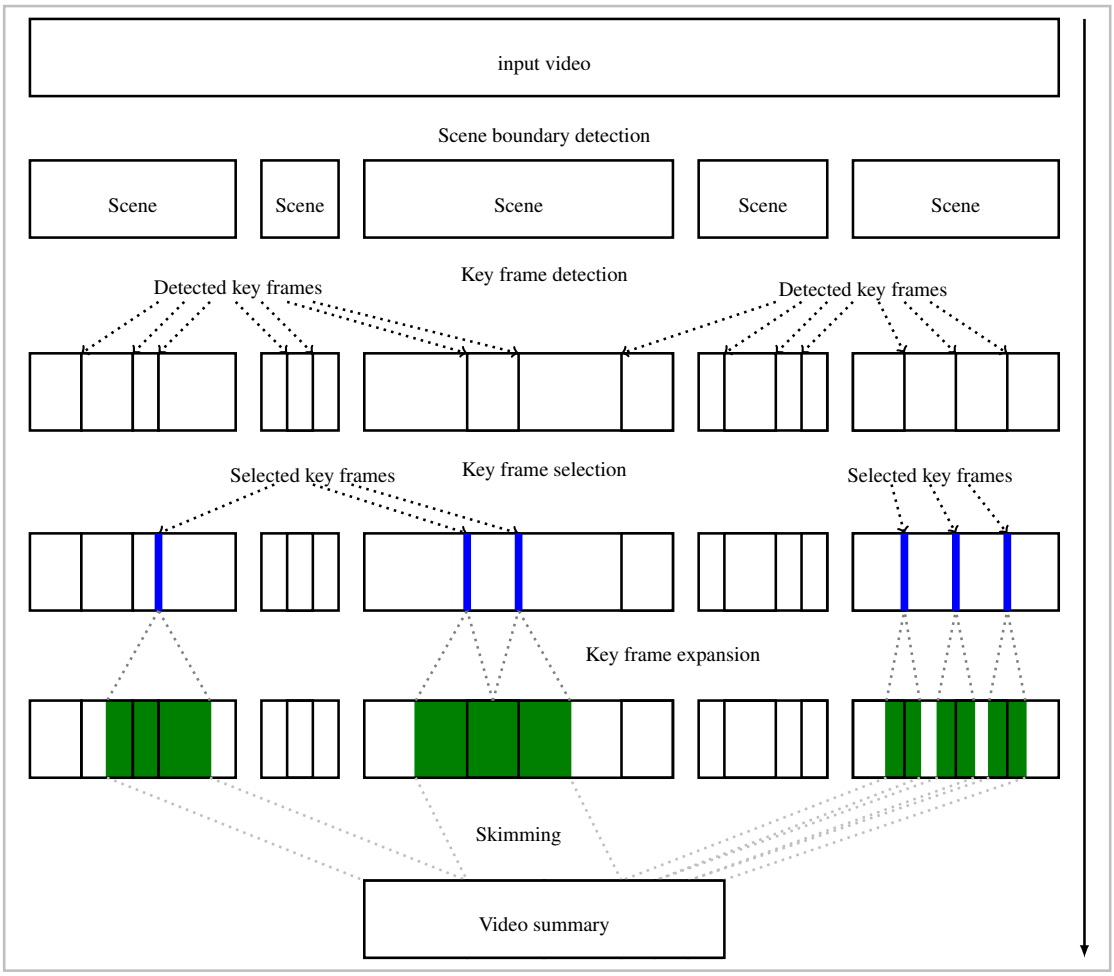


Figure 5. An illustration of the video summarization process in this work.

2.5 Scene boundary detection

Scene boundary detection is a process of analysing a for semantic and structural changes in an attempt to determine where a video scene ends and another starts. For this purpose, a dynamic bag-of-words (BoW) method with dense sampling and scale invariant feature transform (SIFT) detector, by Hietanen was used [22].

The scene boundary detector uses a sliding window in temporal domain within which visual features are first extracted using specialized SIFT detector by David Lowe [23, 24] using dense sampling. Consept of sliding windows is portrayed in Figure 6. Dense sampling means that features are extracted on a regular grid using fixed pixel interval both horizontally and vertically. A codebook of features is then formed from the features found from the initial window. A bag-of-words feature matcher is then used to find features that match the codebook in each window, and matches are distributed into histograms. The histogram acquired from the initial window is used as a comparison histogram. Histograms from consecutive windows are compared to the initial one and if difference is bigger than a given threshold value, a scene boundary is detected. When boundary is detected the window is used as initial one and the process stars over. A flowchart of the boundary detection process is seen in Figure 7. [22]

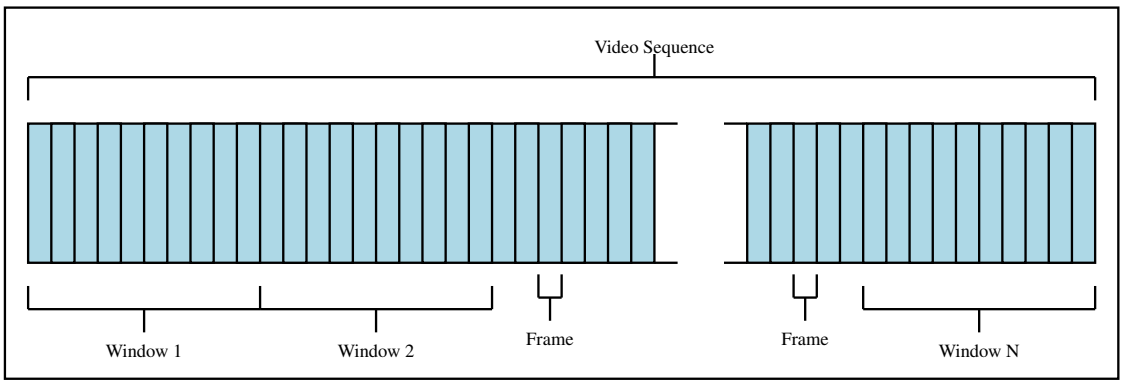


Figure 6. Flow chart of the scene boundary detection algorithm.

Ideally the algorithm works online. This means that further processing could be started before the whole video has gone through the boundary detection process. Because of sliding temporal window, the accuracy of the location of the boundaries is limited. The actual boundary is located somewhere within the last window before the detected boundary location, but the method cannot tell the exact location. This is a minor issue, because the frame level accuracy on cut location would only be achievable when no transitional effects are used and because temporal windows have short durations of one to five seconds.

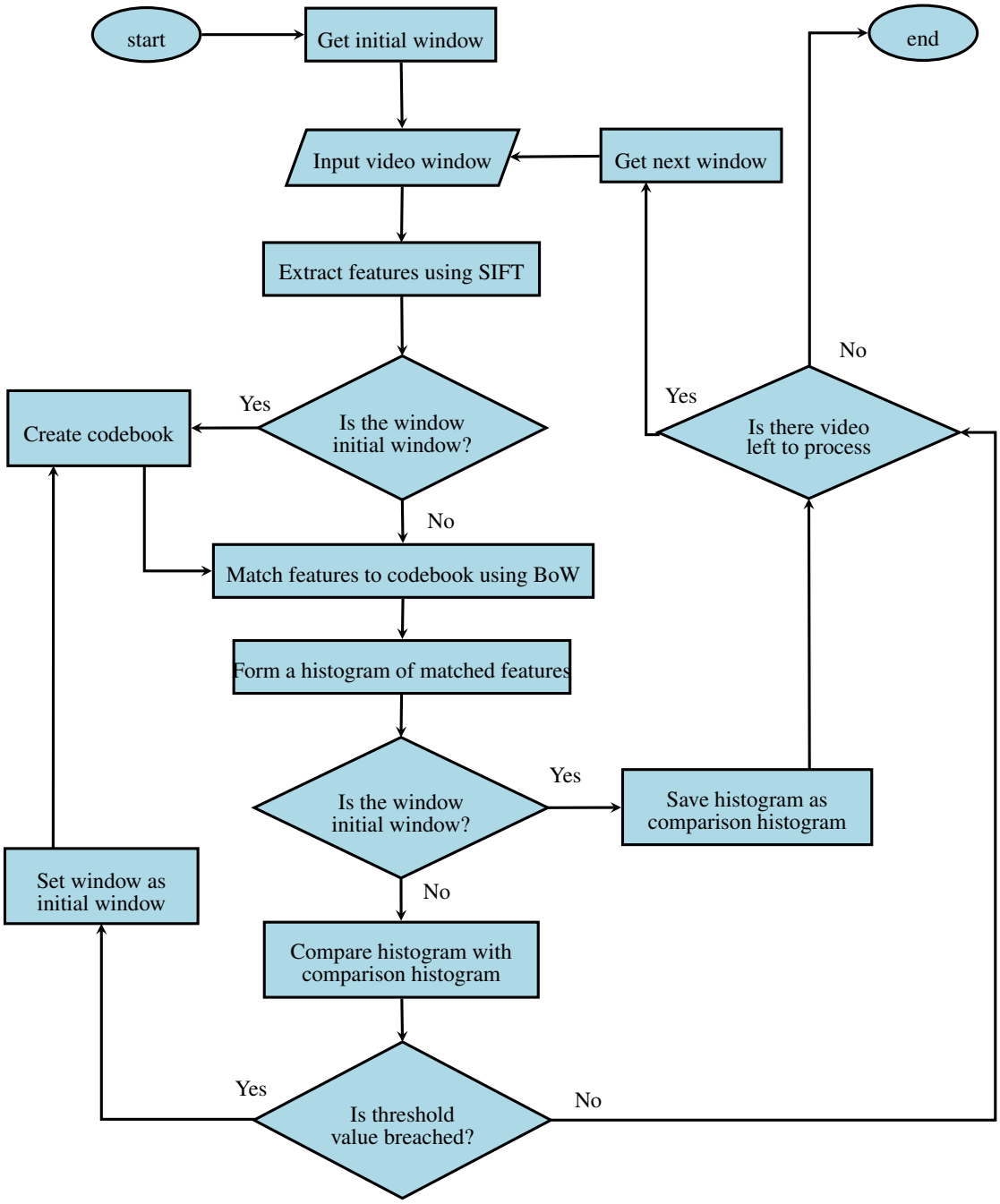


Figure 7. Flow chart of the scene boundary detection algorithm.

2.6 Key frame detection

In key frame detection, a video sequence is analysed to find a frame, or a small set of frames, to represent the video sequence. The detection serves two purposes; Finding the frame which best describes the overall content of the scene and temporal location around which the most interesting things in the scene is happening.

In its simplest form key frame detection is done by uniform sampling. The key frames are selected to be every Nth frame from the video. This however produces variable amounts of frames. If N is set to be higher than the amount of frames in scene, no frames may get picked. If N is too low, too many frames may be selected to sensibly present to the viewer. More analysis is needed in order to get a constant amount of key frames that better represents the video content.

2.6.1 Motion Analysis

An analysis of different video summarization techniques by Ajmal et al. [10] show that motion based approach is suitable for most types of videos. In moving pictures industry, actors and directors use movement to emphasize importance of the moment. Directors pause or or slow down camera pans and actors freeze their gestures to give attention on key moments of scenes [11]. This suggests that measuring the amount of movement can be used to identify the key frames.

On the other hand when filming an even in an anticipation of something happening, most of the footage may be static and the interesting part of the video would have the most motion. In general it can be assumed that the most important moments in the videos are when something is happening and when something stops happening. In case the video is one big blur of an event, such as filming a bike ride with egocentric action camera, it would be good to see what, on average, is happening in the video. Without more sophisticated methods to interpret the semantic contents of the video, it is impossible to know which is the case.

A few simple methods were chosen and compared to test their performance. Farnebäck's dense optical flow [25] was chosen a basis of these analysis. Farnebäck's dense optical flow estimates the motion between image pair by measuring change in position of pixels between then. Dense optical flow measures the amount of movement for every pixel. The magnitude and direction of the motion of each pixel between the frames is given as a

result. A visualization of optical flow in video frame is presented in Figure 8.

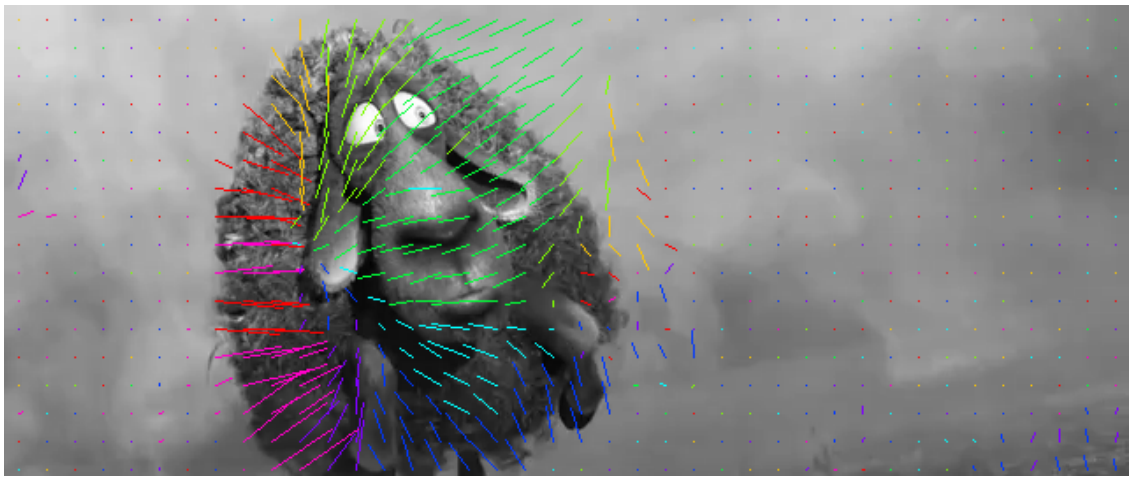


Figure 8. Optical flow on video frame from Cosmos Laundromat[5]. Fot visualization purposes, the motion vectors have been sub-sampled to show only every 16th vector horizontally and vertically.

The minimum, maximum and average motion within a scene can be estimated simply by using the magnitude of flow of each frame, but in order to take into account the direction of the movement, more complex methods are needed. In this work, an approach using direction histograms was used. In this approach, the magnitudes of optical flow of a frame are distributed in histogram bins based on their direction according to Algorithm 1. Figure 9 shows a directional histogram corresponding to the flow seen in Figure 8.

Algorithm 1 Distribution of optical flow to direction histograms

1: I = the number of histogram bins
2: H = array of histogram bins with size I
3: **for all** optical flow vectors with magnitude v and direction θ **do**
4: $\quad i = \left\lfloor I \frac{\theta}{2\pi} \right\rfloor$
5: $\quad H$ at $i = H$ at $i + v$
6: **end for**

2.6.2 Direction histogram comparison

Unlike the total magnitude, the directions do not have a clear minimum and maximum. Thus, the analysis of the direction histograms was done by finding frames having maximum and minimum deviances from a constant reference histogram. The reference histograms were created by finding the average motion in each direction over the scene being

[5]Cosmos Laundromat, directed by Mathieu Auvray (2015; Blender Foundation).

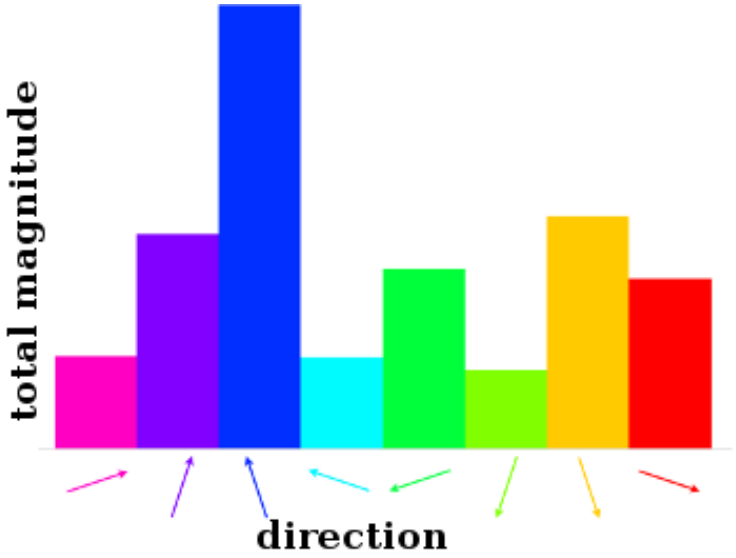


Figure 9. Optical flow directional histogram.

processed. Mean and median functions were used to find averages. The benefits of using mean average function is that it is very easy and fast to compute. Median may however yield better results as it suppresses the effects of extremely low and high peak values. This means that if a video has a short period of movement which significantly differs from rest of the content, e.g., someone bumps into the camera shaking it, or a corrupt frame, the content does not affect the average value as much as it would when using mean.

The performance of L2 norm, chi-square distance, Bhattacharyya distance, correlation and intersection for comparing the histograms were evaluated. The math behind the comparison functions is given in Section 3.4.3, because some of the definitions for the comparison functions depend on their implementation.

L2 norm, or euclidean distance, is a simple way to tell how much difference there is between two histograms, or in this work the difference in the motion. Using chi-square distance, the distance is weighted so that it is relational to the overall magnitude. The amount of difference is considered more meaningful when magnitudes are low. Bhattacharyya distance weights histogram differences similarly, but the weighing is done based on mean value of all histogram bins rather than per bin basis as in chi-square distance.

The comparisons using L2 norm, chi-square distance, Bhattacharyya distance and correlation retain the information about magnitude. Intersection comparison does not. Intersection function takes the smaller value of each corresponding bins in the compared histograms. This makes no sense unless histograms are normalised. When histograms are normalised the intersection tells if the histogram shapes are similar, or if the move-

ment in compared frames, regardless of the speed, is going towards similar direction. The comparisons and their results are presented in Chapter 4.

2.7 Graphical user interface

Video material is far from homogeneous. Some videos may contain fast movement throughout while some are static. Important bits may be happening when something happens in generally static video or something stops in generally high entropy video.

Finding what is the important bit is not straightforward, as an image that would seem having a high importance based on composition and overall image quality might occur in a middle of a scene by accident; A person may be seen in a frame as in posing for the shot, but frame is actually just a part of a longer pan shot of a larger object, and the person filming has no interest whatsoever on the solitary person in image. Creating a good video summary is a subjective problem. People may find different things interesting, depending on their needs.

It is very difficult for a simple automatic system to differentiate the semantic content. It is even more difficult to predict the user's needs and preferences based on the video content. These problems are tackled by creating an easy to use user interface which allows for human annotations. Such interface enables the user to discard content that the key frame detection algorithms have falsely identified as relevant. The user can also hand pick content that fits his needs at the time with minimum effort.

3 IMPLEMENTATION

The implemented video summarization workflow is presented in Figure 10. The input video is first preprocessed by transcoding it to a more manageable size in order to cut down processing time and requirements for temporary data storage. The preprocessed intermediate video is fed into the scene boundary detector, which returns information about scene boundaries. The boundary information is then used to split input video in scenes and key frame detection algorithms find key frames for each scene. Key frames are presented to user in a storyboard view. user can select the key frames which are the most interesting or descriptive and the parts of video to be use in skimming are picked around the selected key frames according to given summary target length.

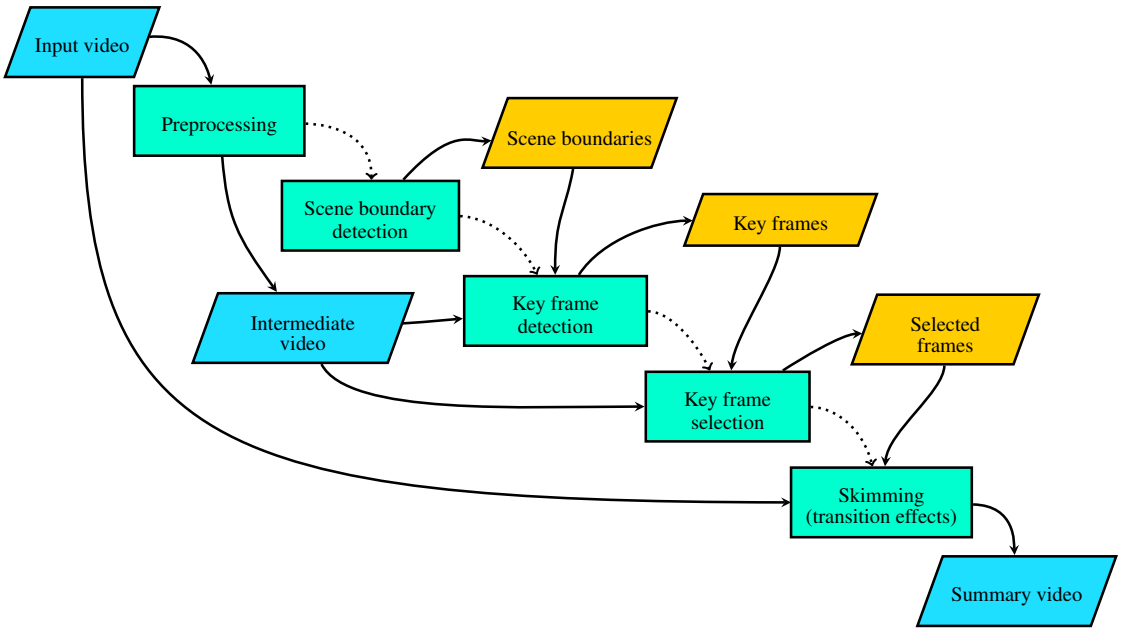


Figure 10. Video summarization workflow used in this work.

3.1 Tools and software libraries

The implementation of the software relies heavily on tools and libraries created by others. The video encoding during preprocessing and final skimming is done using FFmpeg [26]. FFmpeg is a free versatile framework for multimedia conversion. It is cross-platform compatible and is capable of processing a wide variety of common multimedia formats. The framework provides command line tools and development libraries for viewing, encoding and transcoding video and audio.

The key frame detection algorithm uses algorithms implemented in Open Source Computer Vision Library, better known as OpenCV [27]. OpenCV library contains optimized computer vision algorithms for a many video and image processing tasks. It has a large user base and algorithms implemented are well tested. For this reason it was also used in this work.

Graphical User interface was created using QtWebKitWidgets-library. The library is a part of Qt [28] user interface development framework and it provides an API for using QTWebKit [29], which is a WebKit [30] based engine for rendering HTML, CSS and JavaScript content. The combination of HTML, CSS and JavaScript make it possible to create the user interface with relative ease, and making it possible to port the user interface to a web service in the future. In order to use web content in desktop application, a HTML, CSS and JavaScript rendering engine must be used. The QtWebKit was chosen, because it seemed to be easier to be integrated using the QtWebKitWidgets-library, than the WebKit engine it is based on.

3.2 Preprocessing and intermediate video format

In preprocessing the input videos were scaled to width of 360 pixels. Re-encoding was done using H264 video codec, which is an efficient video compression standard [31]. This format has been found to be suitable as it retains good enough quality and level of detail for processing while reducing the amount of data so that it can be processed with moderate time and resources. Encoding was done using FFmpeg (version 2.8.6) command line tool with NVENC [32]. NVENC makes it possible to use specialized hardware of compatible graphics cards for encoding supported codecs, which makes the process faster. The options used for encoding without frame rate conversion were

```
-vf scale=360:-2 -c:v nvenc_h264 -an .
```

In order to re-encode with framerate conversion, options used were

```
-r N -vf scale=360:-2 -c:v nvenc_h264 -an ,
```

where output video frame rate is denoted by N. Constant frame rates of 15 and 24 frames per second were used in the experiments.

For the set of videos used in the tests the average compression ratio against the original H264-encoded videos was 17.4 % without framerate conversion. The compression ratio was 11.2 % when encoded to constant framerate of 15 fps and using 24 fps it was 15.6 %.

3.3 Scene boundary detection

The scene boundary detection was done using implementation by Hietanen [22], described briefly in Chapter 2. The parameters used for boundary detection were:

Window size: Equivalent of one second.
Number of features: 1000
Codebook size: 50
Threshold: 0.54

These values were found to perform well without requiring too much computational resources. The windows size equivalent of one second performs well at detecting the cuts, even with slow transitions, but being short enough so that short shots or scenes are not left completely undetected. With 1000 features used the boundary detection performs well, but using fewer would cause the result to degrade. [22]

3.4 Key frame detection

Most of the implementation work done was focused on the software for key frame detection. The implementation was programmed using C++ with OpevCV -library (version 2.4.8). The implemented key frame detection algorithms were uniform sampling and motion analysis.

3.4.1 Uniform sampling

Uniform sampling, every Nth frame is selected as a key frame, which results in variable amounts of frames selected when scene lengths differ. In order to overcome this problem, the uniform sampling was implemented using a scene length normalization. Each scene was set to have a length of 1, and sampling was done using interval of 0.5. In short, a middle frame from each scene was selected as key frame.

3.4.2 SIFT features

The scene boundary detector described in Section 2.5 uses SIFT feature histograms to detect scene boundary locations based on dissimilarity peaks. An illustration of the dissimilarity is presented in Figure 11. Some of the dissimilarity peaks occur within the scenes, but when a threshold value is not breach they are not considered scene changes. The location of these peaks are easily made available for later process with minimum amount of additional computation. For this reason the usefulness of the available precomputed information in key frame detection was tested.

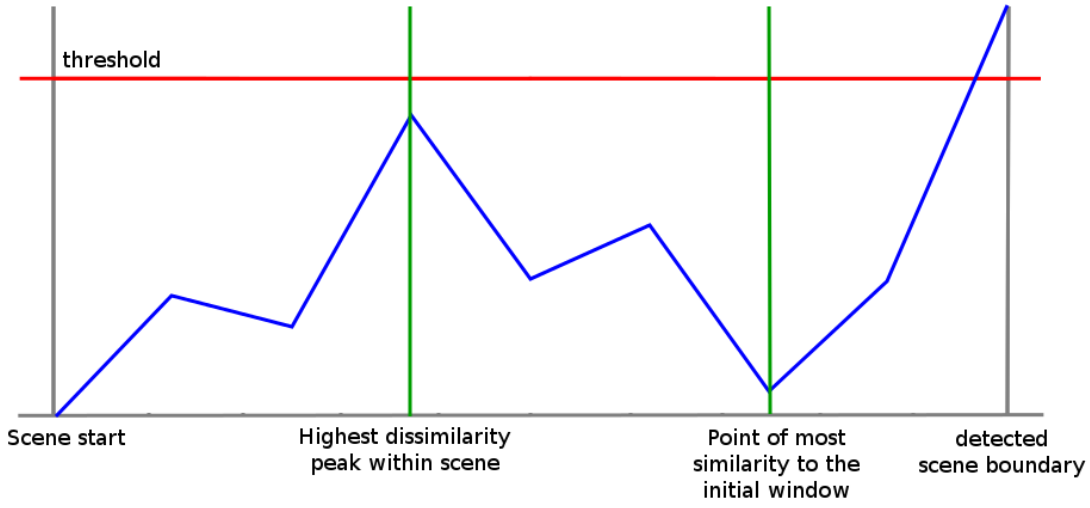


Figure 11. Dissimilarity (blue) within a detected scene.

The scene boundary detector was modified so that in addition to the scene boundary locations, it produces the location of the highest dissimilarity peak before the scene boundary and the location of the most similarity to the initial frame window (see Figure 11). The first frames of produced windows were then then used as key frames.

3.4.3 Motion analysis

The motion analysis implemented was based on OpenCV implementation of Farnebäck's dense optical flow. Optical flow describes difference between video frames by measuring change in position of pixels between images. Dense optical flow measures the amount of movement for every pixel in image pair. [25]

The scene boundary detection algorithm is only as accurate as the window size. This means that the actual boundary location is not known, but it is somewhere within the last second before the marked frame. This inaccuracy has to be taken into account. During a cut the Farnebäck's optical flow algorithm is likely to return disproportionally large movement during a hard cut. This would result in the frame after the cut to have a very high distance from the average frame of the scene being evaluated and thus a cut is likely to be detected instead of a relevant key frame. The problem is illustrated in Figure 12. The problem with inaccuracy was circumvented by disregarding the last second of each scene during motion analysis.

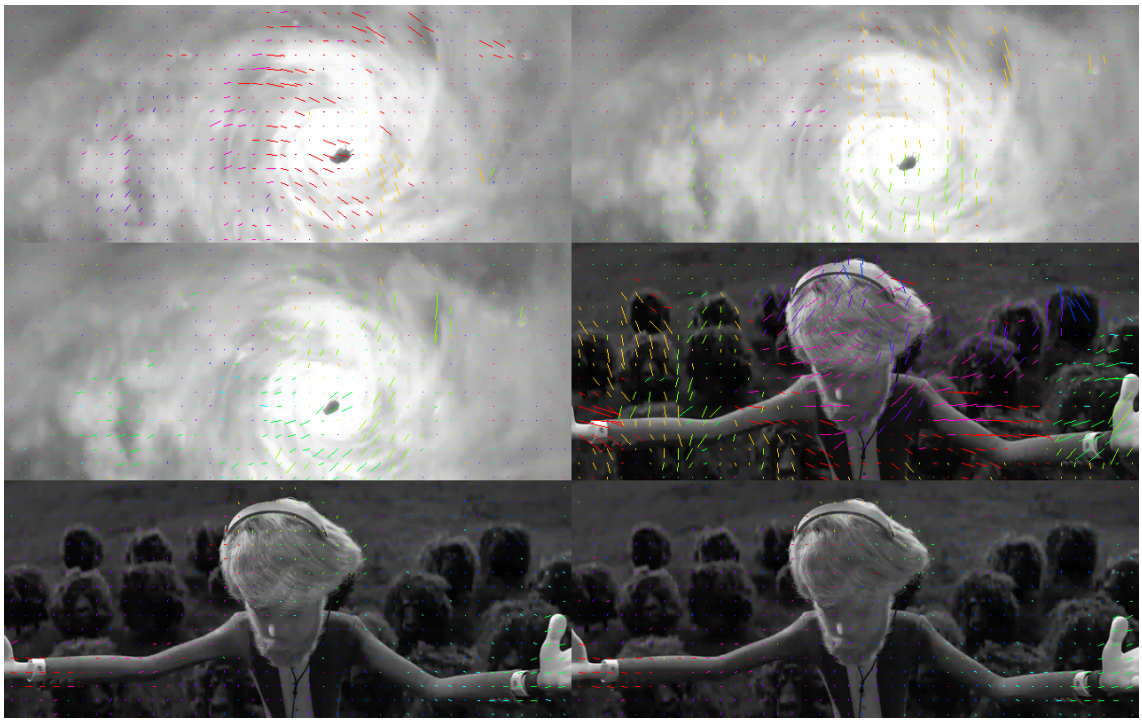


Figure 12. Motion vectors acquired by Farnebäck's optical flow on frames during a hard cut. The frame right after the cut shows a lot of motion.

Taking into account the common instructions used for the spatial composition, it may make some sense to focus the motion analysis around the center of the frames and around the areas 33.3 % and 38.2 % from the edges of the frame, as these are the areas where the objects filmed are likely to be located. The focusing was implemented by weighting the magnitude components around this areas as illustrated in Figure 13. The magnitudes were weighted by 10 % when the pixels were located within the 25 % to 50 % from the edge of the frame or by 20 % if located within 30 % to 40 % or 45 % to 50 % from the edge of the frame. This weighing was made in both horizontal and vertical directions separately.

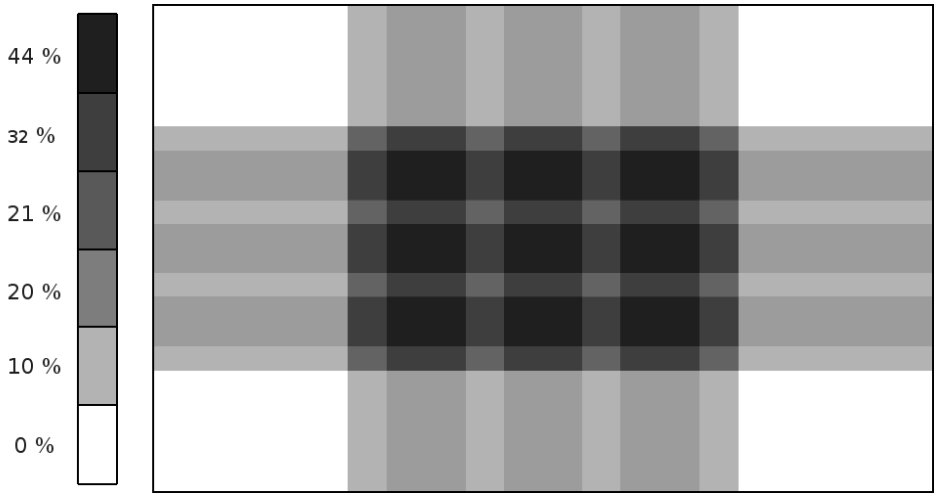


Figure 13. An illustration of spatial weight grid based on common guides for image composition. The darker color denotes higher weight.

Minimum, maximum and average motion

For minimum, maximum and mean flows the directional information was discarded and the evaluation was based only on total amount of movement between frames. This was computed as total magnitude, V, of flow in each frame according to (1)

$$V = \sum_{x=1}^{w} \sum_{y=1}^{h} v(x, y), \tag{1}$$

where w is the width, h is the height and v is the magnitude component of a motion vector. The minima was computed according to (2)

$$V_{min} = \min_{t \in S} V(t), S := \{1..N\} \tag{2}$$

and maxima according to (3)

$$V_{max} = \max_{t \in S} V(t), S := \{1..N\}, \tag{3}$$

where N is the number of frames in the video.

Motion histogram comparisons

For the motion histogram comparisons, the optical flow vectors were distributed to 8 bins according to their direction as described in Algorithm 1. A mean histogram was computed by getting the mean values of each histogram bin according to (4)

$$H_{mean}(i) = \frac{\sum_{n=0}^{N} H_n(i)}{N}, \tag{4}$$

where i is the index of the histogram bin, n is the frame number and N is the total number of frames. In order to create a median histogram, the histograms were stored in a matrix so that each row of the matrix was one histogram. The matrix columns were then sorted and median histogram was formed according to (5)

$$H_{median}(i) = \begin{cases} \frac{M((N/2-1),i)+M((N/2),i)}{2} & : N \bmod 2 = 0 \\ M(\frac{N+1}{2}, i) & : N \bmod 2 = 1 \end{cases}, \tag{5}$$

where M is the matrix with sorted columns, i is the index of the histogram bin and N is the total number of rows in the matrix.

Using the OpenCV implementation of chi-square function (6) [33],

$$\chi^2(H_{avg}, H_{frame}) = \sum_{n=1}^{N} \frac{(H_{avg}(n) - H_{frame}(n))^2}{H_{avg}(n)}, \tag{6}$$

the motion in average histogram is used as a weighing component. This treats the motion in scenes with little movement as more important than the movement in scenes with a lot of motion. In Bhattacharyya distance (7) [33],

$$d(H_{avg}, H_{frame}) = \sqrt{1 - \frac{1}{\sqrt{\bar{H}_{avg}\bar{H}_{frame}}} \sum_{n=1}^{N} \sqrt{H_{avg}(n)H_{frame}(n)}}, \tag{7}$$

the weighing is done according to mean motion in both average histogram and the histogram of the frame being compared highlighting the the changes in motion with low

magnitudes.

In order to do histogram comparisons by intersection, the histograms were locally L1 normalized according to (8)

$$H_{norm}(i) = \frac{H(i)}{\sum_{n=0}^{N} H(n)}, \tag{8}$$

so that the sum of the histogram bins is always equal to 1.

The intersection was then computed by (9).

$$D = \sum_{n=0}^{N} min(H(n), H_{avg}(n)) \tag{9}$$

3.5 Graphical user interface

The initial idea of the user interface and the human involvement in the summarization process was to select only one key frame for each scene. Users would then be to select which scenes they want included in the summary and which they do not. A screen shot of the first prototype of the graphical user interface is seen in Figure 14.

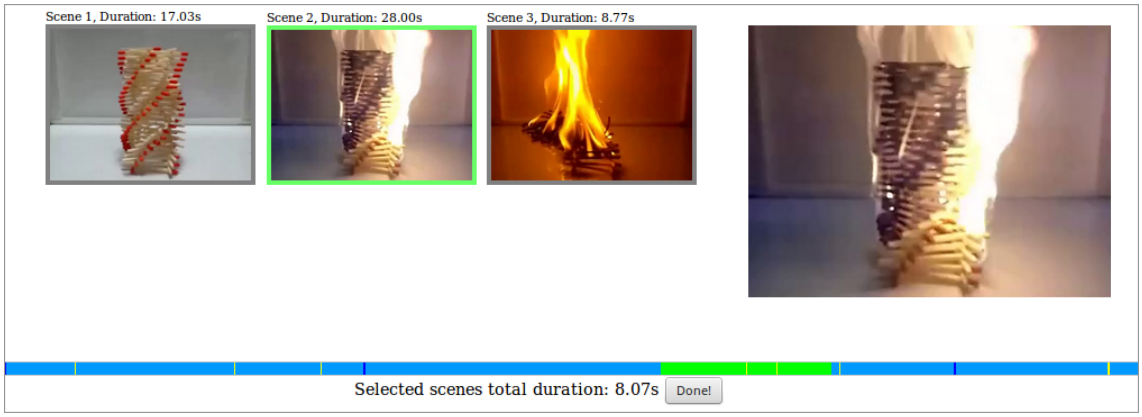


Figure 14. The initial graphical user interface for selecting scenes. Each frame represent different consequent scene detected by the scene boundary detector.

As the key frame detection methods were tested, it became evident that no single method

performs superiorly on all videos. Therefore three methods for key frame detection were then selected to be used with graphical user interface, according to their performance with videos filmed with moving, static and egocentric cameras. More details on the evaluation and results are discussed in Section 4.3. As seen in Figure 15, with three automatically selected frames per scene the user interface does not get too cluttered while giving the user sufficient choices for picking the relevant content with minimal effort.

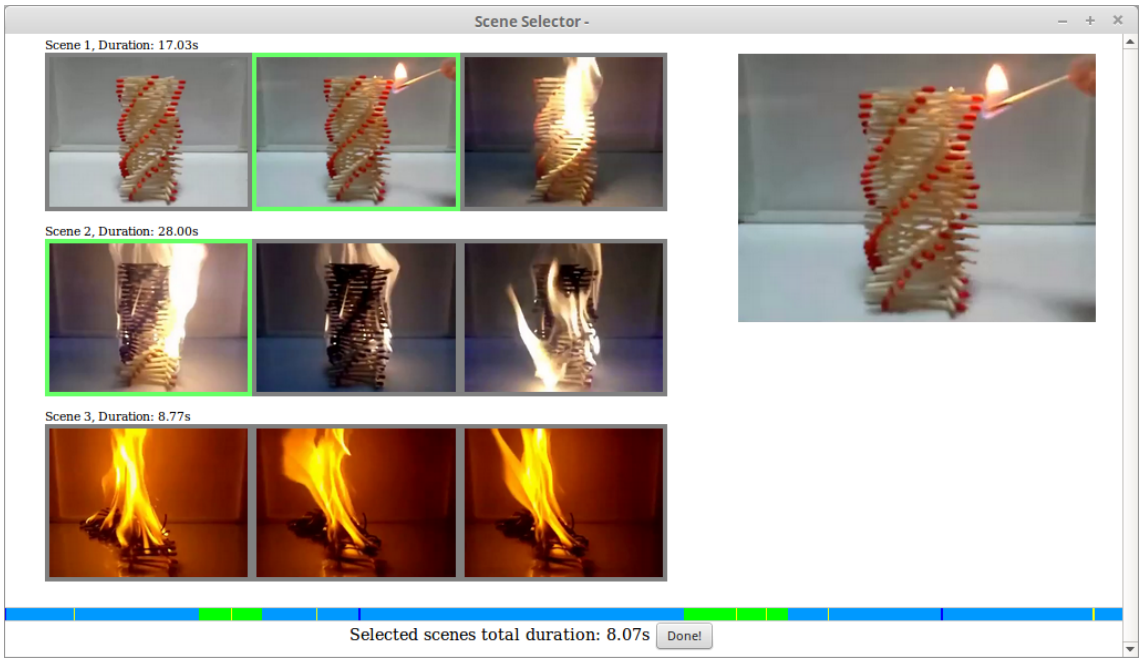


Figure 15. Graphical user interface for selecting key frames produced by motion analysis. Each row represent different consequent scene detected by the scene boundary detector.

Each row of images in the user interface represent different consequent scenes detected by the scene boundary detector. The images are the key frames found using motion analysis in chronological order. In addition to the detected key frames, the user interface displays the duration of each scene to help users select relevant content. The total duration of the selected scenes and a graphical representation of the selections made in relation to the original content is seen as a timeline at the bottom. A slightly larger preview of a key frame can be viewed by single clicking an image. A selection is done by double-clicking an image and selected key frames are highlighted with green border.

3.6 Skimming

To produce the summarized video skim the key frame information needs to be transformed back into a video sequence. This is done by creating shots around the selected key frames and then combining the shots into a single consecutive video.

At first the total duration of the scenes which have key frames selected is determined. If the the total duration is shorter than the target duration of the summary, the scenes are included in the summary entirely. If the total duration is longer than target duration, the scenes need to be shortened in order to meet the target. In this case each scene with selected key frames is shortened proportionally to their length. The maximum duration of each scene in skim, t_{max}, is determined by

$$t_{max} = t_{target} \frac{t_{scene}}{t_{total}}, \tag{10}$$

where t_{target} is the target summary length, t_{scene} is the duration of the scene and t_{total} is the combined duration of all the scenes with selected key frames. This way the length of the parts included are determined by the duration of the scenes in input video. The longer the scenes are in input video, the longer portions of them will be included to resulting skims. The scenes are then split into shots. The number of shots is initially determined by the number of selected key frames within a scene, but if the key frames are close to each other they may be combined in order to avoid overlapping. An illustration of such event is presented in Figure 16.

The start and end frames of the shots within a scene are found according to Algorithm 2. The algorithm first allocates the time reserved for the scene around the selected frames creating shots with uniform length. It then ensures that each shot stays within the boundaries of the selected scenes. After this, the shots are scanned for overlapping. In the case there is overlapping the shots are combined keeping the time allocated to them constant, and again ensuring the combined shots stay within scene boundaries. Once all the selected scenes are processed, the final skim is created by combining all the shots into a single video.

The final skim is created with ffmpeg using the determined shot boundaries by re-encoding the original unprocessed input video. This ensures maximum quality on the end result regardless of the intermediate processing of the video.

Algorithm 2 Finding start and end frames of shots for skims. Shots are clips of video formed around the selected key frames

Clear array of shots
$delta$ = frame rate * (t_{max}/number of selected key frames in selected scene)/2
// find the initial start and end frames of the shots
for all selected key frames in selected scene **do**
    first frame of the shot = selected key frame number - $delta$
    last frame of the shot = selected key frame number + $delta$
    CHECK BOUNDARIES(selected scene, shot)
    Add shot to array of shots.
end for
// Go through shots within the scene and handle overlapping.
while number of shots in array of shots > 1 **do**
    if last frame of first shot in array of shots > first frame of second shot in array of shots **then**
        delta = (last frame of first shot in array of shots - first frame of second shot in array of shots)/2
        first frame of first shot in array of shots = first frame of first shot in array of shots - delta
        last frame of first shot in array of shots = last frame of second shot in array of shots + delta
        CHECK BOUNDARIES(selected scene, shot)
        remove second shot from array of shots
    end if
end while

// Function for ensuring shots stay within scene boundaries
function CHECK BOUNDARIES(scene, shot)
    if first frame of shot < first frame of scene **then**
        last frame of shot = last frame of shot + (first frame of scene - first frame of shot);
        first frame of shot = first frame of scene
    else if last frame of shot > last frame of scene **then**
        first frame of shot = first frame of shot - (last frame of shot - last frame of scene);
        last frame of shot = last frame of scene
    end if
end function

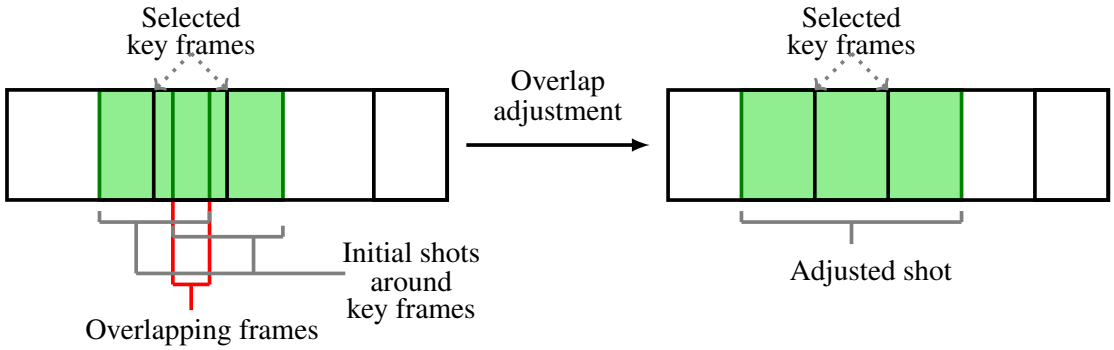


Figure 16. Overlapping of shots formed around key frames that are located near each other. Overlapping shots are combined into a single shot.

4 EXPERIMENTS

4.1 Data

Gygli et al.[14] have introduced a video summary benchmark, called SumMe, to objectively test and compare the performance of video summarization algorithms. The proposed benchmark uses a predefined set of short raw footage video clips. It consists of 25 videos, of which four are filmed using egocentric, four static and 17 using moving cameras. Examples of the videos are seen in Figures 17, 18 and 19. The videos are 30 to 240 seconds long with combined total duration of one hour, six minutes and 18 seconds.

Figure 17. A video of downhill biking recorded with an egocentric helmet camera at Valparaiso .

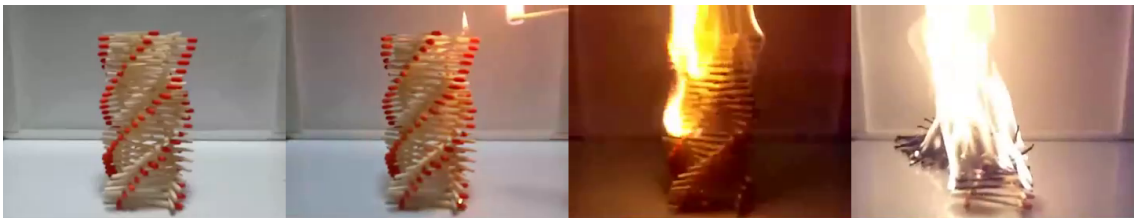


Figure 18. A matchstick tower is set on fire on a video recorded with static camera.

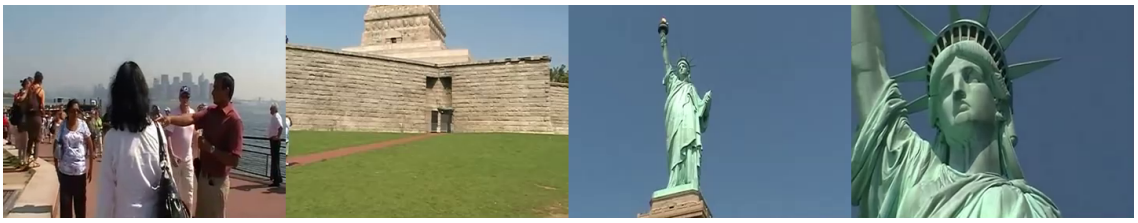


Figure 19. A panning and zooming video of Statue of Liberty recorded with moving camera on Liberty Island.

The ground truth summaries were created by asking human test subjects to summarize a set of test videos so that they would retain the most important content. The summaries length was from 5 % to 15 % of the original videos. These summaries are used as a ground

truth data for performance. The performance of a summarization algorithm is determined by the quality of the summaries it produces. The quality is computed as an average of per-frame pairwise f-measure (11)

$$F_s = \frac{1}{N} \sum_{i=1}^{N} 2 \frac{p_{is} r_{is}}{p_{is} + r_{is}}, \tag{11}$$

where N is the number of ground truth summaries, p is the precision and r the recall of the summary s being evaluated. The precision p is computed according to (12)

$$p = \frac{|n_{gt} \cup n_s|}{|n_s|} \tag{12}$$

and recall r is computed according to (13)

$$r = \frac{|n_{gt} \cup n_s|}{|n_{gt}|}, \tag{13}$$

where n_{gt} is the number of frames in ground truth summary and n_s the number of frames in the summary being evaluated. SumMe benchmark aims at providing an automatic and quantitative way of ranking summarization algorithms, that requires less manual labour than human evaluations. [14]

In this work the SumMe benchmark and the accompanying set of videos were used to test performance of the variations of key frame detection algorithm. The tests were performed by creating summaries with target length of 15 % of the original.

4.2 Human annotations

In order to test the performance of the method with user selectable key frames, the prototype user interface described in Section 3.5 was ported to a web application with precomputed key frames for each of the test videos. The scene boundary detector found from 1 to 34 scenes from the test videos, 9 scenes on average. A screenshot of the user interface with video where multiple scenes were detected is presented in Figure 20 and screenshot when only one scene was found is presented in Figure 21.

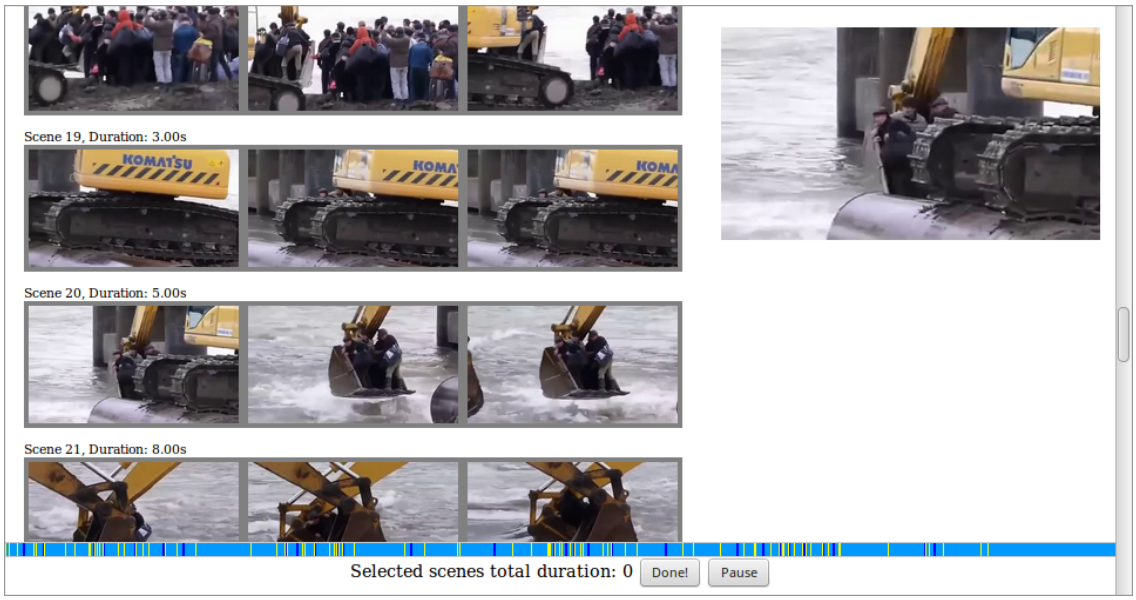


Figure 20. A screenshot of the graphical user interface with video consisting of multiple detected scenes.

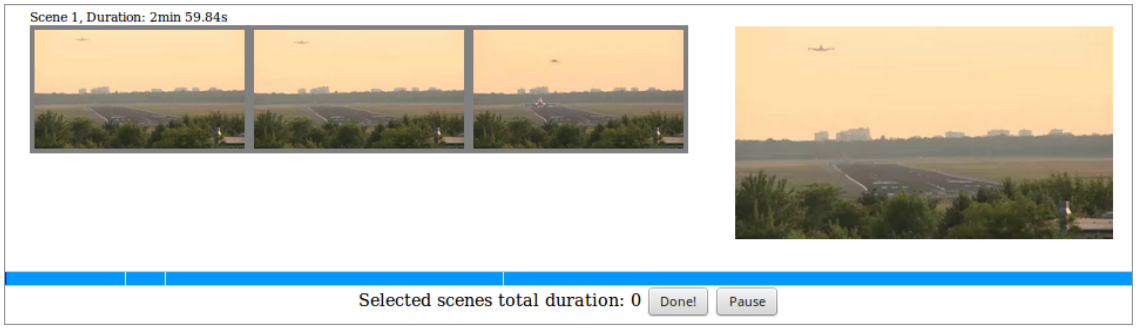


Figure 21. A screenshot of the graphical user interface with video consisting of one detected scene.

4 to 5 test subjects annotated the test video set using the application. The test subjects were of various backgrounds and had no expertise on video production. They were assumed to have no prior knowledge of the contents of the videos. The videos were presented in predefined order and subjects were asked to go through all the videos, preferably on one sitting. The annotations were assumed to take 10 to 25 minutes per person for the whole set.

The annotations were done with the same 15 % target length as the automated tests. The selection of the frames was not limited, allowing the resulting summaries to become even shorter if too many scenes were discarded.

4.3 Results and analysis

4.3.1 Comparison of key frame detection methods

In order to determine which key frame detection method would have the best performance, they were used in unsupervised summarization inthefirst experiment. One key frame was detected from each scene. Frames were then extracted around each of the detected frames to form a skim. The SumMe benchmark was then used to evaluate the performance of the methods.

The first tested methods for finding the key frames were picking the middle frames of the scenes and using the SIFT feature dissimilarity information obtained during scene boundary detection. The tests were performed using multiple frame rates in order to see if frame rate conversions during preprocessing has notable effect on the end results. The full results of the benchmark tests are included in Table A1 of Appendix A. Figure 22 shows a graph of the results compared to performances of SumMe method, mean human and random generated summarization.

Based on the tests the performance of methods using the SIFT feature dissimilarity and picking the middle frames as key frames are comparable to selecting frames at random to create a summary. The performance was equally poor regardless of the frame rates used in preprocessing videos.

Next, methods based on motion analysis were tested. The complete results are included in Appendix B. Using only total magnitude of optical flow, the key frames were detected based on the minimum, maximum, mean and median motion. Spatial weighing of the op-

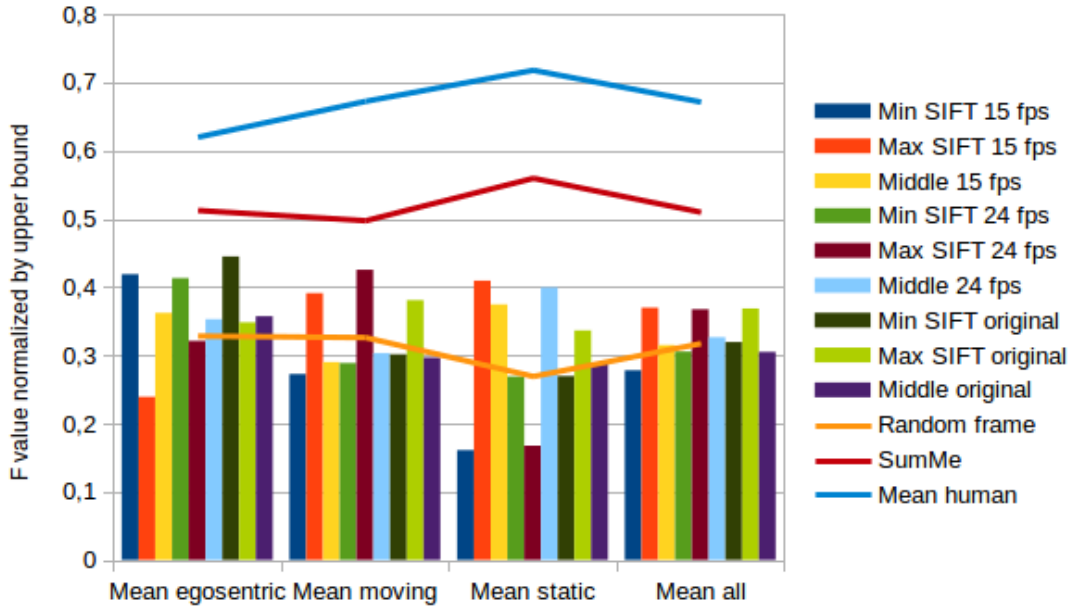


Figure 22. Comparison of the performance of using a single middle frame and SIFT feature dissimilarity method for key frame selection with different frame rates.

tical flow was experimented at this point. Table B2 in Appendix B includes these results. Comparison of the methods with and without spatial weighing shown in Figure 23 shows that spatial weighting did not improve the resulting summaries. A possible for this is that the videos are made by non-professionals who may not be aware of the commonly used guidelines for video scene composition. For this reason spatial weighing was disabled during the consequent tests.

The comparison of the results for single methods show that even when some methods performed well on some videos, none of the methods were superior on all the videos. Figure 24 shows the comparison of the best performing motion analysis based methods when original frame rates of the videos were retained. For videos using moving cameras the best methods were minimum optical flow and maximum deviance from mean flow using intersection as histogram comparison function. For static cameras the best performing methods were maximum deviances from mean and median using L2 norm, and minimum deviance from mean using chi-square distance and minimum distance from median using intersection for egocentric cameras. As seen in Figure 24, methods that performed well on videos recorded with static cameras did not fare as well with videos recorded using egocentric cameras and vice versa.

Due to the performance issues with single key frame detection methods, a few were chosen and combined for the human annotations. The selection of the methods to be combined was made using a simple ranking system. The ranking was used in order to reduce

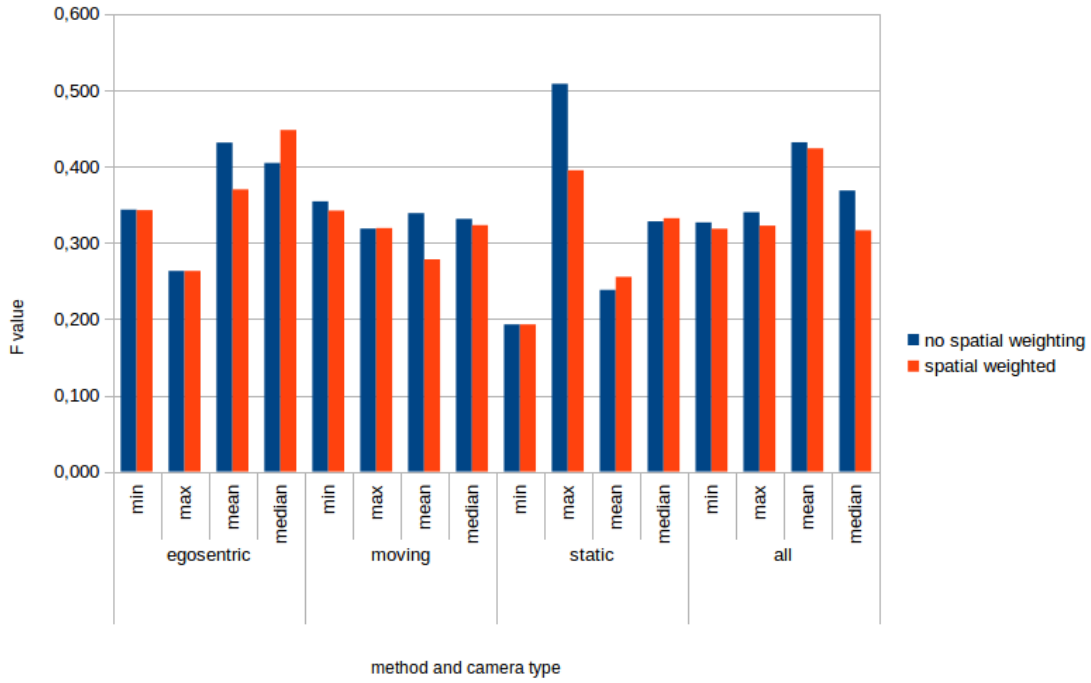


Figure 23. Comparison of the performance of motion analysis based methods.

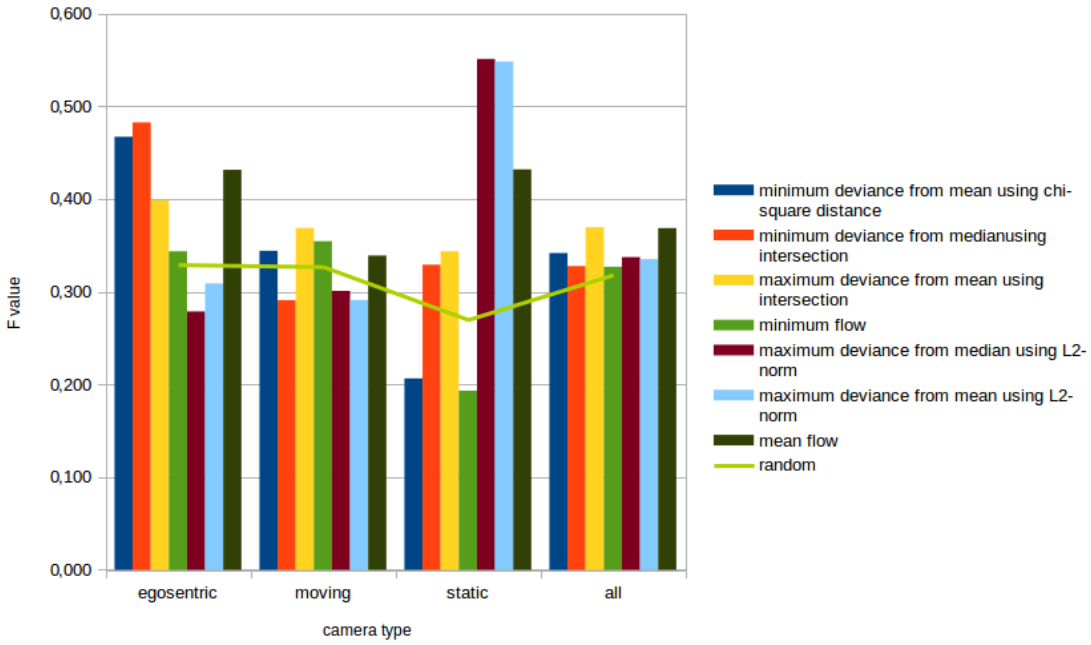


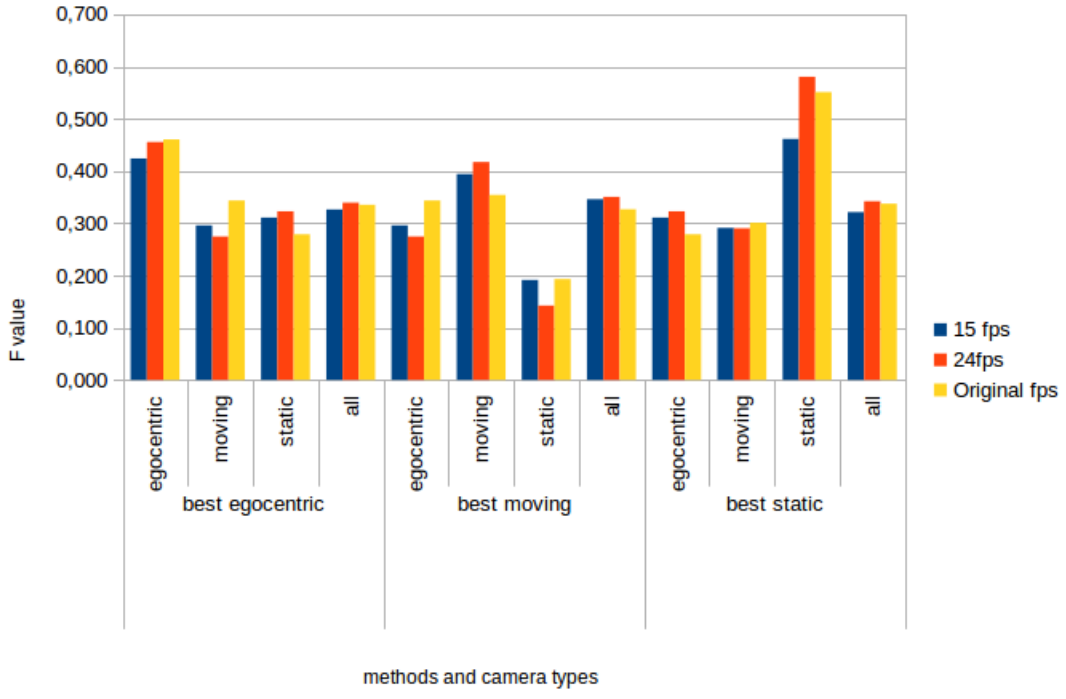
Figure 24. Comparison of the performance of motion analysis based methods.

correlation between the selected methods so that there would be variety in the detected frames. The ranking was done by taking the mean F value obtained using the SumMe benchmark for each camera type and dividing it by the sum of the F values for the other camera types.

The methods which were ranked highest varied depending on the frame rate conversion done during preprocessing. The highest ranking methods with differents framerates are presented in Table 1. A comparison of the highest ranking methods is graphed in Figure 25. Using different frame rates did not yield significant differences in the end results.

Table 1. Highest ranked methods for videos recorded with different types of cameras when using various frame rates.

fps \ camera type	egosentric	moving	static
15	minimum correlation with median histogram	minimum flow	minimum L2 norm from mean histogram
24	minimum intersection with median histogram	minimum correlation with median histogram	maximum L2 norm to median histogram
original	minimum Bhattacharyya distance to mean histogram	minimum flow	maximum L2 norm to median histogram



a

Figure 25. Comparison of the performance of the highest ranking unsupervised methods for each camera type using various frame rates.

Using original frame rates simplifies the preprocessing step and the composition of the final skim, but it also affects the number of frames that need to be processed. More computation resources are naturally needed for processing a higher number of video frames. It is however unclear whether the amount of reduced frames outweighs the simpler transcoding process, and thus original frame rates were used for the human annotation tests.

4.3.2 Comparison to human annotations

The human users were shown key frames for each scene. The frames were those that had the minimum motion, minimum Bhattacharyya distance from mean histogram and maximum L2 norm from median histogram. The benchmark results for the semi-supervized summaries are included in Appendix C. The time span taken for the test subjects to annotate the key frames was a little wider than estimated. The mean time taken for the annotations by the human test subjects was 17 minutes and 36 seconds, fastest being 5 minutes and 59 seconds and slowest being 34 minutes and 59 seconds.

As expected, human annotation brought improvement on the summaries. As illustrated in Figure 26, the human annotations were on par with the best unsupervised method on videos recorded using static cameras and they surpassed all the automatic methods on videos recorded using moving cameras. The egocentric videos were found to be more difficult.

The performance seems poor especially with egocentric videos, even though only one of the summaries was evaluated as particularly bad. This video contains blueish green tinted underwater footage by a scuba diver. Most of the video is imagery of corals which, atleast to a layman, seem homogeneous, and some close ups to various fish. Even though they are quite obvious when looking at the video, the fish, which may be the interesting content of the video, are difficult to see when looking at still images, such as the key frame in Figure 27.

Another example of a poor performance was found with a video where a camera was attached to a wing of a small radio controlled aircraft during a flight. Each key frame extracted from this video is masked by the aircraft and as seen in Figure 28, apart from some changes in the background scenery, all the frames looked very much alike. Unless some users find cloud formations or other elements in the background particularly interesting, the frame selection becomes essentially a guesswork.

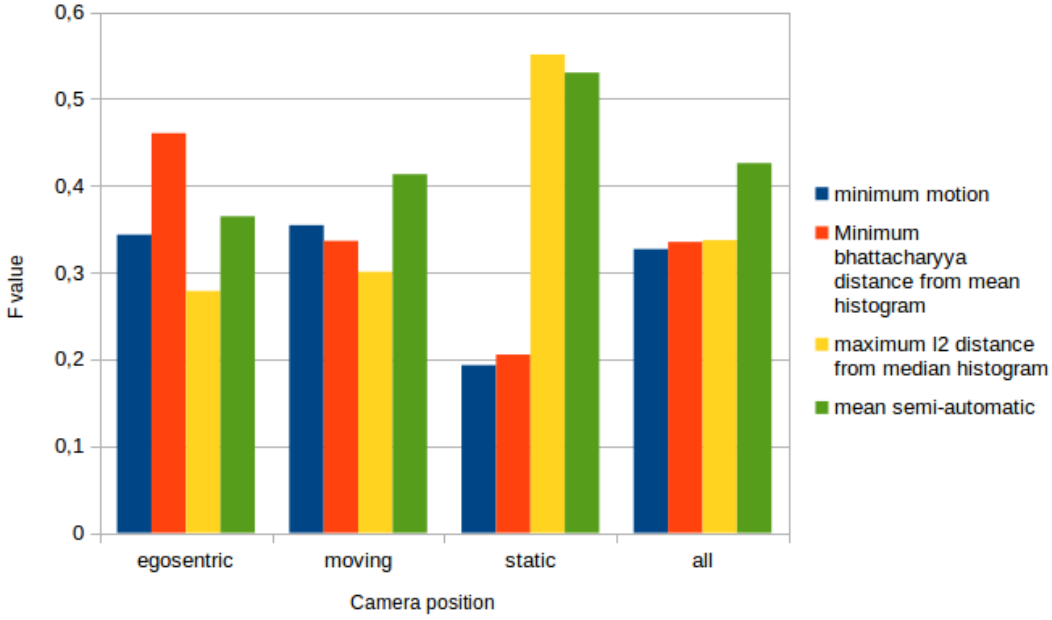


Figure 26. Comparison of the performance of motion analysis based unsupervised methods and combined semi-supervised method.

Figure 27. Fish, located inside the red box, are difficult to be seen in a screen shot of a video.

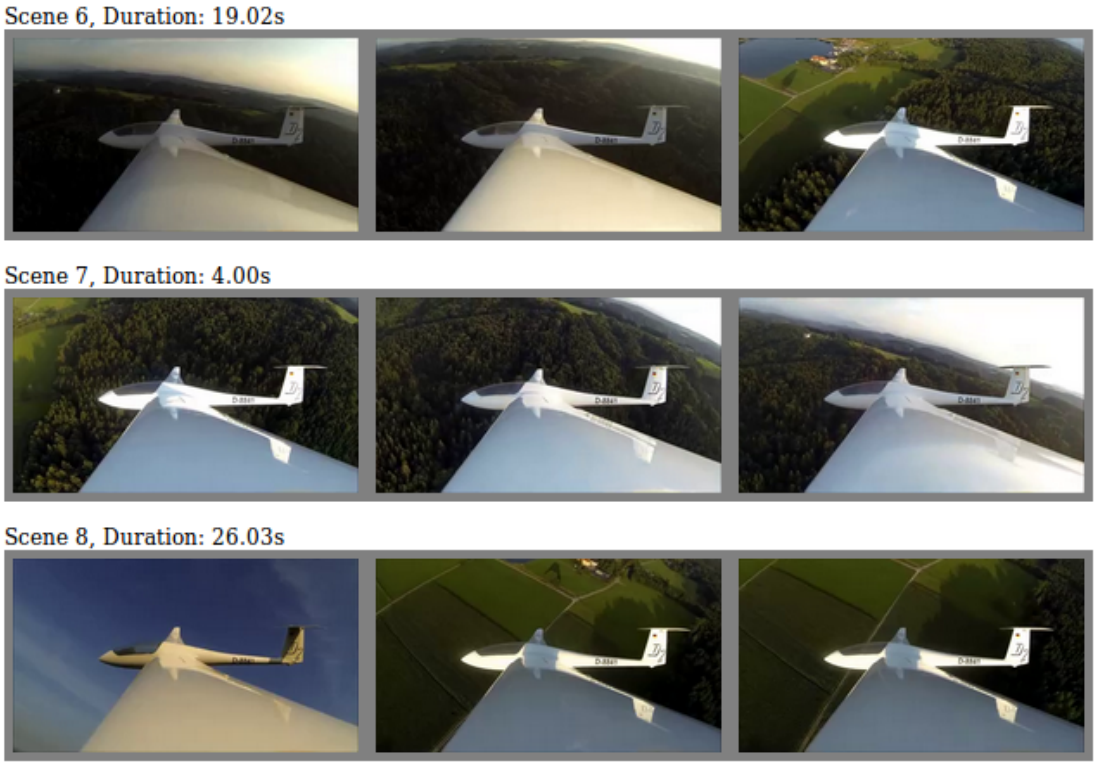


Figure 28. Key frames in a video masked by an aircraft.

On some videos the key frame extraction clearly failed to detect the action. In Figure 29 are the key frames extracted from a video in which persons jump into a pond from rocks. It is not seen when the divers, seen on the background, jump because their movement is dominated by that of the spectators and waterfall on the foreground. In Figure 30 are the key frames extracted from a video where a school of dolphins swim shore and they are subsequently rescued by a group of people. No dolphins nor their rescue operation is seen on the key frames. Both these videos are recorded by hand held devices and they are shaky. The shakiness may cause problems for the motion analysis. Some motion compensation prior the analysis might help achieving better results in these cases.

Figure 29. Dives are not seen in the detected key frames.

Figure 30. No dolphins nor their rescue operation is seen in the detected key frames.

A positive example of the performance of the video summarization method can be seen in a video where a landing of an aircraft has been recorded with a static camera. The key frame detection method is able to pick the moment where the landing gear hits the ground. The key frame is shown in Figure 31. It could however be argued, that the method detected the movement of the bird which happened to fly by the camera at the moment of the plane landing.

Figure 31. Key frame, on which an aircraft lands[6].

Overall the method performed adequately. When comparing to SumMe-method, the semi-automated approach implemented in this work does not yield as good results, but it outperforms an automated approach based on visual attention. This is illustrated in Figure 32.

Some usability issues arose with the human test subjects. Even with instructions, the users sometimes tried to use single clicks instead of double clicks to select frames. This resulted in skims that had no frames selected at all and in these cases the benchmark results were ignored.

[6] `https://www.vision.ee.ethz.ch/~gyglim/vsum/index.php#benchmark`

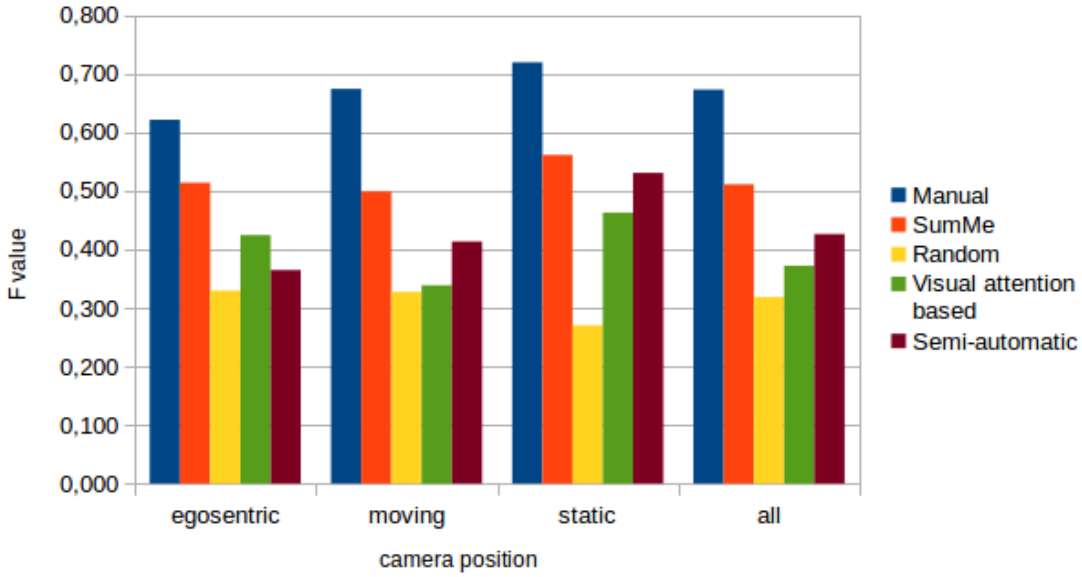


Figure 32. Comparison of the performance of the implemented semi-automatic method against SumMe, manual, visual attention based and random generation methods. The results on SumMe, manual, visual attention based and random generation methods were evaluated by Gygli et al. [14]

5 DISCUSSION AND FUTURE WORK

5.1 Test methodology

Some questions arose during the testing and implementation of the summarization workflow. The validity of the benchmark used may be questionable. The video dataset is quite small. When divided into subcategories based on the camera types used to record them, the small dataset size becomes even more prominent. Only 4 video samples may not be comprehensive enough to provide reliable results. All the videos in the test video set are also quite short; average length being 2 minutes and 39 seconds with longest video having duration of 6 minutes. Longer videos allow more variation within the content and people may find different points of interest. This raises a question about how feasible an objective benchmark is when evaluating subjective matters.

Tests done using larger set of videos, that contains longer recordings would be beneficial. Extending the benchmark by finding suitable videos and enough test subjects to manually summarize them would however require too much time and resources to be included in this thesis. Subjective tests having people summarize the videos using the tool and then evaluating the results could also be beneficial.

The fact that test subjects had no prior knowledge of the contents of the videos in the test set may also affect the end results. If used on home videos users themselves have recorded with a vision of the end result, the knowledge of recorded content would probably be helpful when trying to identify and select the key frames with most relevant content.

5.2 Further development

To develop the summarization workflow further, several improvements can be made. Are the parameters for the scene boundary detection optimal for all videos? The parameters used had been found to be working well on most videos but perhaps they be tuned further. Could the parameters be found dynamically to adjust in various video content? Maybe the scene boundary detector could be tweaked to process the last window of each detected scene further in order to improve accuracy.

The key frame detection implemented in this work was done using dense optical flow, where motion estimation was done for every pixel in a frame. Using sparse flow could

reduce the data, and time used processing it. This may require some normalization, if the amount of motion vectors does not remain the same throughout the scenes. Use of some motion compensation algorithm prior to motion analysis may improve the performance in shaky scenes.

It is quite possible that the motion analysis is not the best possible method for detecting key frames. There are a lot of different methods for key frame detection. Are they better, simpler or faster than the one implemented? YouTube thumbnailer for example finds good thumbnail images from videos using deep neural networks [34]. Something similar could work for key frame detection as well, if suitable training sets are available.

In order to create more entertaining and useful summaries time needed for artistic transitional effects could be taken into account. Perhaps a minimum length for a shot should be defined in order to eliminate a possibility of creating fast blinking shots one or two frames long.

The graphical user interface could be improved as well. As noted in Section 4.3, it is sometimes difficult to see what is going on in the video using a still frame. Replacing the larger preview image with a few frames long video preview might ease this problem.

6 CONCLUSIONS

An approach to summarize home videos by extracting key frames using motion analysis and creating skims around the key frames was implemented and tested. The motion analysis was first tested creating skims automatically. Based on the evaluation of the automatically generated summaries, the best methods were chosen to be used for selecting key frames in semi-automatic version of the approach. For the semi-automated version selected key frames were presented to users, who could choose which frames would best depict the content of the videos and their own interests. Summaries created using this approach were again evaluated.

Although the approaches implemented did not fare as well as the state of the art method, it was shown that decent results could be achieved even using very simple methods and with minimum amount of human effort. It was also shown that the human input is useful in interpreting the semantic contents and recognizing the relevant parts of the videos. The test result may need to be taken with a bit of salt as the benchmark used for the tests had only a few videos on certain categories, and the test data consisted only of very short clips.

The usefulness of the implemented approach for editing longer videos should be investigated further. This however is very time consuming, not only because the video processing is time consuming, but also because the the subjective nature of the editing process.

REFERENCES

[1] Investor's Business Daily. (2014, Nov) GoPro Domin ates Do-It-Yourself Action Video Industry. Http://www.nasdaq.com/article/gopro-dominates-do-it-yourself-action-video-industry-cm414301 (Accessed 12 January 2016).

[2] "IFA 2015 Preview: World's first connected steam oven with integrated camera lets you create delicious dishes through your mobile device," Aug 2015, http://newsroom.electrolux.com/uk/2015/08/25/ifa-2015-preview-world%C2%B4s-first-connected-steam-oven-with-integrated-camera-lets-you-create-delicious-dishes-through-your-mobile-device/ (Accessed 5 January 2016).

[3] "VIDCON 2015 keynote by YouTube CEO Susan Vojcicki," Sep 2015, https://www.youtube.com/watch?v=O6JPxCBlBh8 (Accessed 9 January 2016).

[4] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon, "I tube, you tube, everybody tubes: Analyzing the world's largest user generated content video system," in *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC '07. New York, NY, USA: ACM, 2007, pp. 1–14. [Online]. Available: http://doi.acm.org/10.1145/1298306.1298309

[5] C. M. Taskiran, Z. Pizlo, A. Amir, D. Ponceleon, and E. J. Delp, "Automated video program summarization using speech transcripts," *IEEE Transactions on Multimedia*, vol. 8, no. 4, pp. 775–791, Aug 2006.

[6] B. T. Truong and S. Venkatesh, "Video abstraction: A systematic review and classification," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 3, no. 1, Feb. 2007. [Online]. Available: http://doi.acm.org/10.1145/1198302.1198305

[7] G. Jing, Y. Hu, Y. Guo, Y. Yu, and W. Wang, "Content-Aware Video2Comics with Manga-Style Layout," *IEEE Transactions on Multimedia (TMM)*, vol. 17, no. 12, pp. 2122–2133, 2015.

[8] M. Wang, R. Hong, X.-T. Yuan, S. Yan, and T.-S. Chua, "Movie2comics: Towards a lively video content presentation," *Multimedia, IEEE Transactions on*, vol. 14, no. 3, pp. 858–870, June 2012.

[9] W.-I. Hwang, P.-J. Lee, B.-K. Chun, D.-S. Ryu, and H.-G. Cho, "Cinema comics : Cartoon generation from video stream," in *GRAPP 2006 - COMPUTER GRAPHICS THEORY AND APPLICATIONS*, 2006, pp. 299–304. [Online]. Available: http://ncadjarmstrong.com/year-2-media-cultures-print/cinema_comics_video_stream.pdf

[10] M. Ajmal, M. Ashraf, M. Shakir, Y. Abbas, and F. Shah, "Video summarization: Techniques and classification," in *Computer Vision and Graphics*, ser. Lecture Notes in Computer Science, L. Bolc, R. Tadeusiewicz, L. Chmielewski, and K. Wojciechowski, Eds. Springer Berlin Heidelberg, 2012, vol. 7594, pp. 1–13. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-33564-8_1

[11] W. Wolf, "Key frame selection by motion analysis," in *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, vol. 2, May 1996, pp. 1228–1231 vol. 2.

[12] T. Wang and H. Snoussi, "Histograms of optical flow orientation for visual abnormal events detection," in *Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on*, Sept 2012, pp. 13–18.

[13] R. V. H. M. Colque, C. A. C. Júnior, and W. R. Schwartz, "Histograms of optical flow orientation and magnitude to detect anomalous events in videos," in *2015 28th SIBGRAPI Conference on Graphics, Patterns and Images*, Aug 2015, pp. 126–133.

[14] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool, "Creating summaries from user videos," in *Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VII*. Springer International Publishing, 2014, pp. 505–520. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-10584-0_33

[15] N. Ejaz, I. Mehmood, and S. W. Baik, "Efficient visual attention based framework for extracting key frames from videos," *Signal Processing: Image Communication*, vol. 28, no. 1, pp. 34 – 44, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0923596512001828

[16] J. T. Smith, *Remarks on Rural Scenery*, Jun 1797, digitized by the Internet Archive in 2010. [Online]. Available: https://archive.org/details/remarksonruralsc00smit

[17] G. Markowsky, "Misconceptions about the golden ratio," *The College Mathematics Journal*, vol. 23, no. 1, pp. 2–19, 1992. [Online]. Available: http://www.jstor.org/stable/2686193

[18] C. Falbo, "The golden ratio: A contrary viewpoint," *The College Mathematics Journal*, vol. 36, no. 2, pp. 123–134, 2005. [Online]. Available: http://www.jstor.org/stable/30044835

[19] K. Lancaster, *DSLR Cinema: Crafting the Film Look with Large Sensor Video Cameras*, 2nd ed. Focal Press, 2013.

[20] Y. Rui, T. Huang, and S. Mehrotra, "Exploring video structure beyond the shots," in *Multimedia Computing and Systems, 1998. Proceedings. IEEE International Conference on*, Jun 1998, pp. 237–240.

[21] N. Vasconcelos and A. Lippman, "Statistical models of video structure for content analysis and characterization," *Image Processing, IEEE Transactions on*, vol. 9, no. 1, pp. 3–19, Jan 2000.

[22] A. Hietanen, "Local features for visual object matching and video scene detection," Master's thesis, Tampere university of technology, Dec 2015.

[23] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, ser. ICCV '99. Washington, DC, USA: IEEE Computer Society, 1999, pp. 1150–. [Online]. Available: http://dl.acm.org/citation.cfm?id=850924.851523

[24] ——, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004. [Online]. Available: http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94

[25] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *Proceedings of the 13th Scandinavian Conference on Image Analysis*, ser. SCIA'03. Berlin, Heidelberg: Springer-Verlag, 2003, pp. 363–370. [Online]. Available: http://dl.acm.org/citation.cfm?id=1763974.1764031

[26] "About FFmpeg," https://www.ffmpeg.org/about.html (Accessed 10 April 2016).

[27] "About OpenCV," http://opencv.org/about.html (Accessed 10 April 2016).

[28] "Qt | Cross-platform application and UI development framework," 2016, http://www.qt.io/product/ (Accessed 11 April 2016).

[29] "QtWebKit Guide," 2016, http://doc.qt.io/qt-4.8/qtwebkit-guide.html (Accessed 11 April 2016).

[30] "The WebKit Open Source Project," https://webkit.org/project/ (Accessed 11 April 2016).

[31] A. Puri, X. Chen, and A. Luthra, "Video coding using the h.264/mpeg-4 {AVC} compression standard," *Signal Processing: Image Communication*, vol. 19, no. 9, pp. 793 – 849, 2004, technologies enabling Movies on Internet, {HD} DVD, and {DCinema}. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0923596504000566

[32] “NVIDIA Video encoder application note,” 11 2015, http://developer.download.nvidia.com/assets/cuda/files/NVENC_DA-06209-001_v07.pdf (Accessed 10 April 2016).

[33] “OpenCV 2.4.12.0 documentation, Histogram Comparison,” http://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/histogram_comparison/histogram_comparison.html (Accessed 12 April 2016).

[34] V. C. A. t. Weilong Yang, Min-hsuan Tsai and the YouTube Creator team, “Improving YouTube video thumbnails with deep neural nets,” http://googleresearch.blogspot.fi/2015/10/improving-youtube-video-thumbnails-with.html (Accessed 13 April 2016).

APPENDIX A. Summarization benchmark results (SIFT, middle frame)

Table A1. Unsupervised summarization benchmark results for using middle frames and SIFT feature dissimilarity methods for finding key frames.

		SumMe paper				15 FPS			24 FPS			ORIGINAL FPS		
	video	Upper bound	Mean human	SumMe	Random	Min SIFT	Max SIFT	Middle	Min SIFT	Max SIFT	Middle	Min SIFT	Max SIFT	Middle
Egocentric	Base jumping	0,398	0,257	0,121	0,144	0,151	0,095	0,159	0,123	0,086	0,117	0,166	0,076	0,159
	Bike Polo	0,503	0,322	0,356	0,134	0,209	0,099	0,12	0,278	0,103	0,201	0,192	0,099	0,244
	Scuba	0,387	0,217	0,184	0,138	0,172	0,127	0,154	0,171	0,151	0,167	0,15	0,135	0,148
	Valparaiso_Downhill	0,427	0,272	0,242	0,142	0,186	0,082	0,176	0,149	0,202	0,122	0,253	0,28	0,069
Moving	Bearpark_climbing	0,33	0,208	0,118	0,147	0,074	0,108	0,15	0,123	0,147	0,111	0,115	0,111	0,093
	Bus_in_Rock_Tunnel	0,359	0,198	0,135	0,135	0,115	0,185	0,09	0,106	0,188	0,087	0,199	0,183	0,082
	Car_railcrossing	0,515	0,357	0,362	0,14	0,069	0,129	0,072	0,084	0,146	0,064	0,146	0,126	0,082
	Cockpit_Landing	0,443	0,279	0,172	0,136	0,134	0,111	0,115	0,1	0,152	0,138	0,125	0,08	0,143
	Cooking	0,528	0,379	0,321	0,145	0,022	0,253	0,052	0,206	0,188	0,097	0,022	0,253	0,052
	Eiffel Tower	0,467	0,312	0,295	0,13	0,19	0,167	0,146	0,176	0,115	0,166	0,199	0,22	0,144
	Excavators river crossing	0,411	0,303	0,189	0,144	0,129	0,134	0,142	0,142	0,116	0,143	0,161	0,137	0,144
	Jumps	0,611	0,483	0,427	0,149	0,205	0,069	0,096	0,123	0,251	0,091	0,22	0,16	0,101
	Kids_playing_in_leaves	0,394	0,289	0,089	0,139	0,097	0,424	0,069	0,026	0,303	0,085	0,03	0,275	0,085
	Playing_on_water_slide	0,34	0,195	0,2	0,134	0,102	0,151	0,174	0,102	0,145	0,204	0,083	0,075	0,112
	Saving dolphines	0,313	0,188	0,145	0,144	0,16	0,159	0,112	0,111	0,23	0,125	0,108	0,23	0,125
	St Maarten Landing	0,624	0,496	0,313	0,143	0,215	0,098	0,235	0,226	0,203	0,223	0,335	0,121	0,236
	Statue of Liberty	0,332	0,184	0,192	0,122	0,09	0,17	0,115	0,103	0,19	0,08	0,107	0,169	0,082
	Uncut_Evening_Flight	0,506	0,35	0,271	0,131	0,112	0,193	0,072	0,099	0,29	0,078	0,124	0,241	0,108
	paluma_jump	0,662	0,509	0,181	0,139	0,105	0,047	0,281	0,104	0,042	0,281	0,087	0	0,281
	playing_ball	0,403	0,271	0,174	0,145	0,081	0,147	0,071	0,179	0,188	0,132	0,076	0,169	0,217
	Notre_Dame	0,36	0,231	0,235	0,137	0,107	0,185	0,143	0,124	0,146	0,137	0,124	0,146	0,137
static	Air_Force_One	0,49	0,332	0,318	0,144	0,052	0,342	0,279	0,094	0,047	0,28	0,094	0,316	0,28
	Fire Domino	0,514	0,394	0,13	0,145	0,044	0,089	0,141	0,086	0,084	0,191	0,167	0,09	0,143
	car_over_camera	0,49	0,346	0,372	0,134	0,086	0,233	0,236	0,218	0,051	0,239	0,17	0,194	0,067
	Paintball	0,55	0,399	0,32	0,127	0,152	0,16	0,095	0,149	0,168	0,092	0,118	0,071	0,091
		SumMe paper				15 FPS			24 FPS			ORIGINAL FPS		
	video	Upper bound	Mean human	SumMe	Random	Min SIFT	Max SIFT	Middle	Min SIFT	Max SIFT	Middle	Min SIFT	Max SIFT	Middle
Egocentric	Base jumping	1,000	0,646	0,304	0,362	0,379	0,239	0,399	0,309	0,216	0,294	0,417	0,191	0,399
	Bike Polo	1,000	0,640	0,708	0,266	0,416	0,197	0,239	0,553	0,205	0,400	0,382	0,197	0,485
	Scuba	1,000	0,561	0,475	0,357	0,444	0,328	0,398	0,442	0,390	0,432	0,388	0,349	0,382
	Valparaiso_Downhill	1,000	0,637	0,567	0,333	0,436	0,192	0,412	0,349	0,473	0,286	0,593	0,656	0,162
Moving	Bearpark_climbing	1,000	0,630	0,358	0,445	0,224	0,327	0,455	0,373	0,445	0,336	0,348	0,336	0,282
	Bus_in_Rock_Tunnel	1,000	0,552	0,376	0,376	0,320	0,515	0,251	0,295	0,524	0,242	0,554	0,510	0,228
	Car_railcrossing	1,000	0,693	0,703	0,272	0,134	0,250	0,140	0,163	0,283	0,124	0,283	0,245	0,159
	Cockpit_Landing	1,000	0,630	0,388	0,307	0,302	0,251	0,260	0,226	0,343	0,312	0,282	0,181	0,323
	Cooking	1,000	0,718	0,608	0,275	0,042	0,479	0,098	0,390	0,356	0,184	0,042	0,479	0,098
	Eiffel Tower	1,000	0,668	0,632	0,278	0,407	0,358	0,313	0,377	0,246	0,355	0,426	0,471	0,308
	Excavators river crossing	1,000	0,737	0,460	0,350	0,314	0,326	0,345	0,345	0,282	0,348	0,392	0,333	0,350
	Jumps	1,000	0,791	0,699	0,244	0,336	0,113	0,157	0,201	0,411	0,149	0,360	0,262	0,165
	Kids_playing_in_leaves	1,000	0,734	0,226	0,353	0,246	1,076	0,175	0,066	0,769	0,216	0,076	0,698	0,216
	Playing_on_water_slide	1,000	0,574	0,588	0,394	0,300	0,444	0,512	0,300	0,426	0,600	0,244	0,221	0,329
	Saving dolphines	1,000	0,601	0,463	0,460	0,511	0,508	0,358	0,355	0,735	0,399	0,345	0,735	0,399
	St Maarten Landing	1,000	0,795	0,502	0,229	0,345	0,157	0,377	0,362	0,325	0,357	0,537	0,194	0,378
	Statue of Liberty	1,000	0,554	0,578	0,367	0,271	0,512	0,346	0,310	0,572	0,241	0,322	0,509	0,247
	Uncut_Evening_Flight	1,000	0,692	0,536	0,259	0,221	0,381	0,142	0,196	0,573	0,154	0,245	0,476	0,213
	paluma_jump	1,000	0,769	0,273	0,210	0,159	0,071	0,424	0,157	0,063	0,424	0,131	0,000	0,424
	playing_ball	1,000	0,672	0,432	0,360	0,201	0,365	0,176	0,444	0,467	0,328	0,189	0,419	0,538
	Notre_Dame	1,000	0,642	0,653	0,381	0,297	0,514	0,397	0,344	0,406	0,381	0,344	0,406	0,381
static	Air_Force_One	1,000	0,678	0,649	0,294	0,106	0,698	0,569	0,192	0,096	0,571	0,192	0,645	0,571
	Fire Domino	1,000	0,767	0,253	0,282	0,086	0,173	0,274	0,167	0,163	0,372	0,325	0,175	0,278
	car_over_camera	1,000	0,706	0,759	0,273	0,176	0,476	0,482	0,445	0,104	0,488	0,347	0,396	0,137
	Paintball	1,000	0,725	0,582	0,231	0,276	0,291	0,173	0,271	0,305	0,167	0,215	0,129	0,165
Mean	egosentric	1,000	0,621	0,513	0,329	0,419	0,239	0,362	0,413	0,321	0,353	0,445	0,348	0,357
	moving	1,000	0,674	0,498	0,327	0,272	0,391	0,290	0,289	0,425	0,303	0,301	0,381	0,297
	static	1,000	0,719	0,561	0,270	0,161	0,409	0,375	0,269	0,167	0,400	0,270	0,336	0,288
	all	1,000	0,672	0,511	0,318	0,278	0,370	0,315	0,305	0,367	0,326	0,319	0,368	0,305

APPENDIX B. Summarization benchmark results (motion analysis)

Table B1. Summarization benchmark results for methods using minimum, maximum and average optical flows for finding key frames. Using original frame rate in intermediate videos.

										spatial weighted			
	video	Upper bound	Mean human	SumMe	Random	min flow	max flow	mean flow	median flow	min flow	max flow	mean flow	median flow
Egocentric	Base jumping	0,398	0,257	0,121	0,144	0,204	0,105	0,212	0,29	0,203	0,105	0,1	0,091
	Bike Polo	0,503	0,322	0,356	0,134	0,059	0,078	0,112	0,085	0,059	0,078	0,083	0,19
	Scuba	0,387	0,217	0,184	0,138	0,124	0,16	0,177	0,077	0,124	0,16	0,245	0,196
	Valparaiso_Downhill	0,427	0,272	0,242	0,142	0,181	0,094	0,219	0,223	0,181	0,094	0,184	0,29
Moving	Bearpark_climbing	0,33	0,208	0,118	0,147	0,193	0,141	0,264	0,03	0,193	0,142	0,094	0,055
	Bus_in_Rock_Tunnel	0,359	0,198	0,135	0,135	0,114	0,146	0,117	0,155	0,114	0,146	0,201	0,18
	Car_railcrossing	0,515	0,357	0,362	0,14	0,066	0,118	0,285	0,073	0,065	0,118	0,058	0,072
	Cockpit_Landing	0,443	0,279	0,172	0,136	0,122	0,148	0,199	0,135	0,122	0,148	0,085	0,107
	Cooking	0,528	0,379	0,321	0,145	0,316	0,036	0,032	0,144	0,194	0,036	0,093	0,018
	Eiffel Tower	0,467	0,312	0,295	0,13	0,096	0,073	0,096	0,137	0,109	0,073	0,118	0,084
	Excavators river crossing	0,411	0,303	0,189	0,144	0,133	0,138	0,141	0,171	0,133	0,138	0,13	0,185
	Jumps	0,611	0,483	0,427	0,149	0,118	0,133	0,133	0,157	0,118	0,133	0,169	0,173
	Kids_playing_in_leaves	0,394	0,289	0,089	0,139	0,319	0,061	0,117	0,107	0,319	0,061	0,073	0,275
	Playing_on_water_slide	0,34	0,195	0,2	0,134	0,146	0,146	0,192	0,16	0,146	0,146	0,178	0,099
	Saving dolphines	0,313	0,188	0,145	0,144	0,236	0,191	0,113	0,172	0,236	0,191	0,05	0,222
	St Maarten Landing	0,624	0,496	0,313	0,143	0,056	0,099	0,08	0,052	0,056	0,099	0,114	0,041
	Statue of Liberty	0,332	0,184	0,192	0,122	0,117	0,18	0,087	0,186	0,117	0,182	0,142	0,187
	Uncut_Evening_Flight	0,506	0,35	0,271	0,131	0,09	0,231	0,174	0,192	0,09	0,232	0,116	0,094
	paluma_jump	0,662	0,509	0,181	0,139	0,155	0	0,047	0,014	0,155	0	0,085	0,01
	playing_ball	0,403	0,271	0,174	0,145	0,103	0,207	0,167	0,226	0,103	0,207	0,143	0,204
	Notre_Dame	0,36	0,231	0,235	0,137	0,106	0,136	0,131	0,19	0,106	0,136	0,133	0,167
static	Air_Force_One	0,49	0,332	0,318	0,144	0,04	0,322	0,281	0,03	0,04	0,322	0,266	0,05
	Fire Domino	0,514	0,394	0,13	0,145	0,03	0,279	0,185	0,141	0,03	0,046	0,115	0,16
	car_over_camera	0,49	0,346	0,372	0,134	0,121	0,276	0,185	0,111	0,121	0,276	0,197	0,11
	Paintball	0,55	0,399	0,32	0,127	0,212	0,149	0,229	0,215	0,212	0,149	0,29	0,211
		normalized to upper bound											
										spatial weighted			
	video	Upper bound	Mean human	SumMe	Random	min flow	max flow	mean flow	median flow	min flow	max flow	mean flow	median flow
Egocentric	Base jumping	1,000	0,646	0,304	0,362	0,513	0,264	0,533	0,729	0,510	0,264	0,251	0,229
	Bike Polo	1,000	0,640	0,708	0,266	0,117	0,155	0,223	0,169	0,117	0,155	0,165	0,378
	Scuba	1,000	0,561	0,475	0,357	0,320	0,413	0,457	0,199	0,320	0,413	0,633	0,506
	Valparaiso_Downhill	1,000	0,637	0,567	0,333	0,424	0,220	0,513	0,522	0,424	0,220	0,431	0,679
Moving	Bearpark_climbing	1,000	0,630	0,358	0,445	0,585	0,427	0,800	0,091	0,585	0,430	0,285	0,167
	Bus_in_Rock_Tunnel	1,000	0,552	0,376	0,376	0,318	0,407	0,326	0,432	0,318	0,407	0,560	0,501
	Car_railcrossing	1,000	0,693	0,703	0,272	0,128	0,229	0,553	0,142	0,126	0,229	0,113	0,140
	Cockpit_Landing	1,000	0,630	0,388	0,307	0,275	0,334	0,449	0,305	0,275	0,334	0,192	0,242
	Cooking	1,000	0,718	0,608	0,275	0,598	0,068	0,061	0,273	0,367	0,068	0,176	0,034
	Eiffel Tower	1,000	0,668	0,632	0,278	0,206	0,156	0,206	0,293	0,233	0,156	0,253	0,180
	Excavators river crossing	1,000	0,737	0,460	0,350	0,324	0,336	0,343	0,416	0,324	0,336	0,316	0,450
	Jumps	1,000	0,791	0,699	0,244	0,193	0,218	0,218	0,257	0,193	0,218	0,277	0,283
	Kids_playing_in_leaves	1,000	0,734	0,226	0,353	0,810	0,155	0,297	0,272	0,810	0,155	0,185	0,698
	Playing_on_water_slide	1,000	0,574	0,588	0,394	0,429	0,429	0,565	0,471	0,429	0,429	0,524	0,291
	Saving dolphines	1,000	0,601	0,463	0,460	0,754	0,610	0,361	0,550	0,754	0,610	0,160	0,709
	St Maarten Landing	1,000	0,795	0,502	0,229	0,090	0,159	0,128	0,083	0,090	0,159	0,183	0,066
	Statue of Liberty	1,000	0,554	0,578	0,367	0,352	0,542	0,262	0,560	0,352	0,548	0,428	0,563
	Uncut_Evening_Flight	1,000	0,692	0,536	0,259	0,178	0,457	0,344	0,379	0,178	0,458	0,229	0,186
	paluma_jump	1,000	0,769	0,273	0,210	0,234	0,000	0,071	0,021	0,234	0,000	0,128	0,015
	playing_ball	1,000	0,672	0,432	0,360	0,256	0,514	0,414	0,561	0,256	0,514	0,355	0,506
	Notre_Dame	1,000	0,642	0,653	0,381	0,294	0,378	0,364	0,528	0,294	0,378	0,369	0,464
static	Air_Force_One	1,000	0,678	0,649	0,294	0,082	0,657	0,573	0,061	0,082	0,657	0,543	0,102
	Fire Domino	1,000	0,767	0,253	0,282	0,058	0,543	0,360	0,274	0,058	0,089	0,224	0,311
	car_over_camera	1,000	0,706	0,759	0,273	0,247	0,563	0,378	0,227	0,247	0,563	0,402	0,224
	Paintball	1,000	0,725	0,582	0,231	0,385	0,271	0,416	0,391	0,385	0,271	0,527	0,384
Mean	egosentric	1,000	0,621	0,513	0,329	0,344	0,263	0,431	0,405	0,343	0,263	0,370	0,448
	moving	1,000	0,674	0,498	0,327	0,354	0,319	0,339	0,331	0,342	0,319	0,278	0,323
	static	1,000	0,719	0,561	0,270	0,193	0,509	0,432	0,238	0,193	0,395	0,424	0,255
	all	1,000	0,672	0,511	0,318	0,327	0,340	0,369	0,328	0,319	0,323	0,316	0,332

Table B2. Summarization benchmark results for method using motion histogram comparison to detect key frames. Table presents results achieved using different histogram comparison functions with mean average histogram used as comparison histogram. Using original frame rate in intermediate videos.

						mean average histogram									
						max deviance					min deviance				
	video	Upper bound	Mean human	SumMe	Random	bhat	chi	corr	int	l2	bhat	chi	corr	int	l2
Egocentric	Base jumping	0,398	0,257	0,121	0,144	0,09	0,083	0,163	0,118	0,118	0,181	0,205	0,122	0,138	0,179
	Bike Polo	0,503	0,322	0,356	0,134	0,086	0,078	0,081	0,086	0,078	0,118	0,101	0,1	0,122	0,1
	Scuba	0,387	0,217	0,184	0,138	0,163	0,173	0,215	0,166	0,162	0,184	0,184	0,129	0,184	0,203
	Valparaiso_Downhill	0,427	0,272	0,242	0,142	0,298	0,296	0,199	0,298	0,156	0,289	0,289	0,155	0,291	0,084
Moving	Bearpark_climbing	0,33	0,208	0,118	0,147	0,264	0,267	0,173	0,266	0,124	0,164	0,162	0,235	0,162	0,162
	Bus_in_Rock_Tunnel	0,359	0,198	0,135	0,135	0,109	0,12	0,102	0,106	0,146	0,118	0,113	0,121	0,109	0,16
	Car_railcrossing	0,515	0,357	0,362	0,14	0,124	0,13	0,075	0,118	0,121	0,222	0,233	0,185	0,233	0,1
	Cockpit_Landing	0,443	0,279	0,172	0,136	0,095	0,08	0,092	0,142	0,147	0,176	0,179	0,09	0,182	0,223
	Cooking	0,528	0,379	0,321	0,145	0,313	0,318	0,189	0,313	0,034	0,073	0,073	0,131	0,042	0,013
	Eiffel Tower	0,467	0,312	0,295	0,13	0,111	0,139	0,07	0,115	0,182	0,104	0,114	0,164	0,098	0,262
	Excavators river crossing	0,411	0,303	0,189	0,144	0,167	0,169	0,135	0,159	0,134	0,133	0,134	0,132	0,134	0,138
	Jumps	0,611	0,483	0,427	0,149	0,103	0,059	0,149	0,059	0,103	0,256	0,256	0,141	0,256	0,195
	Kids_playing_in_leaves	0,394	0,289	0,089	0,139	0,093	0,084	0,05	0,093	0,068	0,126	0,126	0,096	0,147	0,144
	Playing_on_water_slide	0,34	0,195	0,2	0,134	0,066	0,116	0,155	0,091	0,135	0,173	0,173	0,136	0,173	0,103
	Saving dolphines	0,313	0,188	0,145	0,144	0,291	0,291	0,06	0,291	0,108	0,23	0,23	0,122	0,23	0,122
	St Maarten Landing	0,624	0,496	0,313	0,143	0,278	0,275	0,218	0,234	0,099	0,1	0,089	0,133	0,089	0,108
	Statue of Liberty	0,332	0,184	0,192	0,122	0,091	0,124	0,118	0,12	0,18	0,079	0,082	0,112	0,081	0,115
	Uncut_Evening_Flight	0,506	0,35	0,271	0,131	0,13	0,081	0,163	0,138	0,098	0,098	0,162	0,141	0,137	0,281
	paluma_jump	0,662	0,509	0,181	0,139	0,017	0,047	0,009	0,047	0	0,104	0,104	0,104	0,104	0,112
	playing_ball	0,403	0,271	0,174	0,145	0,136	0,158	0,085	0,186	0,187	0,101	0,101	0,187	0,101	0,1
	Notre_Dame	0,36	0,231	0,235	0,137	0,128	0,104	0,118	0,115	0,136	0,143	0,136	0,162	0,155	0,139
static	Air_Force_One	0,49	0,332	0,318	0,144	0,283	0,283	0,316	0,283	0,37	0,06	0,06	0,047	0,294	0,266
	Fire Domino	0,514	0,394	0,13	0,145	0,169	0,159	0,179	0,156	0,271	0,039	0,037	0,04	0,037	0,133
	car_over_camera	0,49	0,346	0,372	0,134	0,154	0,141	0,173	0,157	0,258	0,241	0,245	0,203	0,248	0,244
	Paintball	0,55	0,399	0,32	0,127	0,139	0,096	0,076	0,095	0,211	0,072	0,072	0,227	0,078	0,102
						normalized to upper bound									
						max deviance					min deviance				
	video	Upper bound	Mean human	SumMe	Random	bhat	chi	corr	int	l2	bhat	chi	corr	int	l2
Egocentric	Base jumping	1,000	0,646	0,304	0,362	0,226	0,209	0,410	0,296	0,296	0,455	0,515	0,307	0,347	0,450
	Bike Polo	1,000	0,640	0,708	0,266	0,171	0,155	0,161	0,171	0,155	0,235	0,201	0,199	0,243	0,199
	Scuba	1,000	0,561	0,475	0,357	0,421	0,447	0,556	0,429	0,419	0,475	0,475	0,333	0,475	0,525
	Valparaiso_Downhill	1,000	0,637	0,567	0,333	0,698	0,693	0,466	0,698	0,365	0,677	0,677	0,363	0,681	0,197
Moving	Bearpark_climbing	1,000	0,630	0,358	0,445	0,800	0,809	0,524	0,806	0,376	0,497	0,491	0,712	0,491	0,491
	Bus_in_Rock_Tunnel	1,000	0,552	0,376	0,376	0,304	0,334	0,284	0,295	0,407	0,329	0,315	0,337	0,304	0,446
	Car_railcrossing	1,000	0,693	0,703	0,272	0,241	0,252	0,146	0,229	0,235	0,431	0,452	0,359	0,452	0,194
	Cockpit_Landing	1,000	0,630	0,388	0,307	0,214	0,181	0,208	0,321	0,332	0,397	0,404	0,203	0,411	0,503
	Cooking	1,000	0,718	0,608	0,275	0,593	0,602	0,358	0,593	0,064	0,138	0,138	0,248	0,080	0,025
	Eiffel Tower	1,000	0,668	0,632	0,278	0,238	0,298	0,150	0,246	0,390	0,223	0,244	0,351	0,210	0,561
	Excavators river crossing	1,000	0,737	0,460	0,350	0,406	0,411	0,328	0,387	0,326	0,324	0,326	0,321	0,326	0,336
	Jumps	1,000	0,791	0,699	0,244	0,169	0,097	0,244	0,097	0,169	0,419	0,419	0,231	0,419	0,319
	Kids_playing_in_leaves	1,000	0,734	0,226	0,353	0,236	0,213	0,127	0,236	0,173	0,320	0,320	0,244	0,373	0,365
	Playing_on_water_slide	1,000	0,574	0,588	0,394	0,194	0,341	0,456	0,268	0,397	0,509	0,509	0,400	0,509	0,303
	Saving dolphines	1,000	0,601	0,463	0,460	0,930	0,930	0,192	0,930	0,345	0,735	0,735	0,390	0,735	0,390
	St Maarten Landing	1,000	0,795	0,502	0,229	0,446	0,441	0,349	0,375	0,159	0,160	0,143	0,213	0,143	0,173
	Statue of Liberty	1,000	0,554	0,578	0,367	0,274	0,373	0,355	0,361	0,542	0,238	0,247	0,337	0,244	0,346
	Uncut_Evening_Flight	1,000	0,692	0,536	0,259	0,257	0,160	0,322	0,273	0,194	0,194	0,320	0,279	0,271	0,555
	paluma_jump	1,000	0,769	0,273	0,210	0,026	0,071	0,014	0,071	0,000	0,157	0,157	0,157	0,157	0,169
	playing_ball	1,000	0,672	0,432	0,360	0,337	0,392	0,211	0,462	0,464	0,251	0,251	0,464	0,251	0,248
	Notre_Dame	1,000	0,642	0,653	0,381	0,356	0,289	0,328	0,319	0,378	0,397	0,378	0,450	0,431	0,386
static	Air_Force_One	1,000	0,678	0,649	0,294	0,578	0,578	0,645	0,578	0,755	0,122	0,122	0,096	0,600	0,543
	Fire Domino	1,000	0,767	0,253	0,282	0,329	0,309	0,348	0,304	0,527	0,076	0,072	0,078	0,072	0,259
	car_over_camera	1,000	0,706	0,759	0,273	0,314	0,288	0,353	0,320	0,527	0,492	0,500	0,414	0,506	0,498
	Paintball	1,000	0,725	0,582	0,231	0,253	0,175	0,138	0,173	0,384	0,131	0,131	0,413	0,142	0,185
Mean	egosentric	1,000	0,621	0,513	0,329	0,379	0,376	0,398	0,399	0,309	0,460	0,467	0,300	0,437	0,342
	moving	1,000	0,674	0,498	0,327	0,354	0,364	0,270	0,369	0,291	0,336	0,344	0,335	0,341	0,342
	static	1,000	0,719	0,561	0,270	0,368	0,337	0,371	0,344	0,548	0,205	0,206	0,250	0,330	0,371
	all	1,000	0,672	0,511	0,318	0,360	0,362	0,307	0,369	0,335	0,335	0,342	0,316	0,355	0,347

Table B3. Summarization benchmark results for method using motion histogram comparison to detect key frames. Table presents results achieved using different histogram comparison functions with median average histogram used as comparison histogram. Using original frame rate in intermediate videos.

						median average histogram									
						max deviance					min deviance				
	video	Upper bound	Mean human	SumMe	Random	bhat	chi	corr	int	l2	bhat	chi	corr	int	l2
Egocentric	Base jumping	0,398	0,257	0,121	0,144	0,123	0,091	0,085	0,121	0,071	0,167	0,161	0,099	0,167	0,055
	Bike Polo	0,503	0,322	0,356	0,134	0,063	0,046	0,166	0,068	0,08	0,132	0,132	0,126	0,218	0,193
	Scuba	0,387	0,217	0,184	0,138	0,165	0,174	0,252	0,137	0,163	0,237	0,202	0,241	0,241	0,123
	Valparaiso_Downhill	0,427	0,272	0,242	0,142	0,296	0,293	0,195	0,219	0,152	0,078	0,078	0,115	0,194	0,14
Moving	Bearpark_climbing	0,33	0,208	0,118	0,147	0,267	0,27	0,203	0,27	0,13	0,06	0,06	0,219	0,06	0,195
	Bus_in_Rock_Tunnel	0,359	0,198	0,135	0,135	0,097	0,117	0,089	0,105	0,146	0,108	0,087	0,114	0,093	0,082
	Car_railcrossing	0,515	0,357	0,362	0,14	0,128	0,109	0,142	0,103	0,121	0,113	0,114	0,086	0,093	0,106
	Cockpit_Landing	0,443	0,279	0,172	0,136	0,097	0,12	0,1	0,097	0,147	0,104	0,079	0,091	0,088	0,214
	Cooking	0,528	0,379	0,321	0,145	0,225	0,023	0,105	0,226	0,034	0,255	0,1	0,136	0,211	0,094
	Eiffel Tower	0,467	0,312	0,295	0,13	0,185	0,192	0,183	0,148	0,234	0,135	0,132	0,159	0,135	0,088
	Excavators river crossing	0,411	0,303	0,189	0,144	0,157	0,136	0,168	0,154	0,138	0,145	0,145	0,126	0,168	0,16
	Jumps	0,611	0,483	0,427	0,149	0,155	0,17	0,122	0,131	0,124	0,255	0,255	0,244	0,271	0,164
	Kids_playing_in_leaves	0,394	0,289	0,089	0,139	0,086	0,09	0,246	0,086	0,068	0,252	0,252	0,253	0,197	0,059
	Playing_on_water_slide	0,34	0,195	0,2	0,134	0,078	0,102	0,11	0,074	0,135	0,18	0,182	0,126	0,174	0,146
	Saving dolphines	0,313	0,188	0,145	0,144	0,291	0,291	0,06	0,291	0,108	0,051	0,051	0,033	0,033	0,23
	St Maarten Landing	0,624	0,496	0,313	0,143	0,241	0,241	0,216	0,241	0,099	0,019	0,02	0,057	0,057	0,032
	Statue of Liberty	0,332	0,184	0,192	0,122	0,096	0,099	0,12	0,089	0,18	0,176	0,165	0,126	0,117	0,074
	Uncut_Evening_Flight	0,506	0,35	0,271	0,131	0,085	0,08	0,165	0,086	0,098	0,089	0,082	0,076	0,086	0,071
	paluma_jump	0,662	0,509	0,181	0,139	0	0,047	0,104	0,047	0	0,017	0,017	0,111	0,184	0,017
	playing_ball	0,403	0,271	0,174	0,145	0,134	0,156	0,131	0,133	0,182	0,087	0,089	0,055	0,087	0,086
	Notre_Dame	0,36	0,231	0,235	0,137	0,123	0,135	0,139	0,11	0,137	0,135	0,135	0,146	0,129	0,115
static	Air_Force_One	0,49	0,332	0,318	0,144	0,051	0,051	0,316	0,051	0,37	0,296	0,296	0,031	0,296	0,045
	Fire Domino	0,514	0,394	0,13	0,145	0,159	0,159	0,023	0,159	0,276	0,142	0,142	0,164	0,142	0,378
	car_over_camera	0,49	0,346	0,372	0,134	0,168	0,228	0,177	0,202	0,258	0,111	0,087	0,157	0,089	0,12
	Paintball	0,55	0,399	0,32	0,127	0,141	0,073	0,14	0,136	0,212	0,135	0,139	0,14	0,14	0,139
		normalized to upper bound													
		max deviance									min deviance				
	video	Upper bound	Mean human	SumMe	Random	bhat	chi	corr	int	l2	bhat	chi	corr	int	l2
Egocentric	Base jumping	1,000	0,646	0,304	0,362	0,309	0,229	0,214	0,304	0,178	0,420	0,405	0,249	0,420	0,138
	Bike Polo	1,000	0,640	0,708	0,266	0,125	0,091	0,330	0,135	0,159	0,262	0,262	0,250	0,433	0,384
	Scuba	1,000	0,561	0,475	0,357	0,426	0,450	0,651	0,354	0,421	0,612	0,522	0,623	0,623	0,318
	Valparaiso_Downhill	1,000	0,637	0,567	0,333	0,693	0,686	0,457	0,513	0,356	0,183	0,183	0,269	0,454	0,328
Moving	Bearpark_climbing	1,000	0,630	0,358	0,445	0,809	0,818	0,615	0,818	0,394	0,182	0,182	0,664	0,182	0,591
	Bus_in_Rock_Tunnel	1,000	0,552	0,376	0,376	0,270	0,326	0,248	0,292	0,407	0,301	0,242	0,318	0,259	0,228
	Car_railcrossing	1,000	0,693	0,703	0,272	0,249	0,212	0,276	0,200	0,235	0,219	0,221	0,167	0,181	0,206
	Cockpit_Landing	1,000	0,630	0,388	0,307	0,219	0,271	0,226	0,219	0,332	0,235	0,178	0,205	0,199	0,483
	Cooking	1,000	0,718	0,608	0,275	0,426	0,044	0,199	0,428	0,064	0,483	0,189	0,258	0,400	0,178
	Eiffel Tower	1,000	0,668	0,632	0,278	0,396	0,411	0,392	0,317	0,501	0,289	0,283	0,340	0,289	0,188
	Excavators river crossing	1,000	0,737	0,460	0,350	0,382	0,331	0,409	0,375	0,336	0,353	0,353	0,307	0,409	0,389
	Jumps	1,000	0,791	0,699	0,244	0,254	0,278	0,200	0,214	0,203	0,417	0,417	0,399	0,444	0,268
	Kids_playing_in_leaves	1,000	0,734	0,226	0,353	0,218	0,228	0,624	0,218	0,173	0,640	0,640	0,642	0,500	0,150
	Playing_on_water_slide	1,000	0,574	0,588	0,394	0,229	0,300	0,324	0,218	0,397	0,529	0,535	0,371	0,512	0,429
	Saving dolphines	1,000	0,601	0,463	0,460	0,930	0,930	0,192	0,930	0,345	0,163	0,163	0,105	0,105	0,735
	St Maarten Landing	1,000	0,795	0,502	0,229	0,386	0,386	0,346	0,386	0,159	0,030	0,032	0,091	0,091	0,051
	Statue of Liberty	1,000	0,554	0,578	0,367	0,289	0,298	0,361	0,268	0,542	0,530	0,497	0,380	0,352	0,223
	Uncut_Evening_Flight	1,000	0,692	0,536	0,259	0,168	0,158	0,326	0,170	0,194	0,176	0,162	0,150	0,170	0,140
	paluma_jump	1,000	0,769	0,273	0,210	0,000	0,071	0,157	0,071	0,000	0,026	0,026	0,168	0,278	0,026
	playing_ball	1,000	0,672	0,432	0,360	0,333	0,387	0,325	0,330	0,452	0,216	0,221	0,136	0,216	0,213
	Notre_Dame	1,000	0,642	0,653	0,381	0,342	0,375	0,386	0,306	0,381	0,375	0,375	0,406	0,358	0,319
static	Air_Force_One	1,000	0,678	0,649	0,294	0,104	0,104	0,645	0,104	0,755	0,604	0,604	0,063	0,604	0,092
	Fire Domino	1,000	0,767	0,253	0,282	0,309	0,309	0,045	0,309	0,537	0,276	0,276	0,319	0,276	0,735
	car_over_camera	1,000	0,706	0,759	0,273	0,343	0,465	0,361	0,412	0,527	0,227	0,178	0,320	0,182	0,245
	Paintball	1,000	0,725	0,582	0,231	0,256	0,133	0,255	0,247	0,385	0,245	0,253	0,255	0,255	0,253
Mean	egosentric	1,000	0,621	0,513	0,329	0,388	0,364	0,413	0,327	0,279	0,369	0,343	0,348	0,483	0,292
	moving	1,000	0,674	0,498	0,327	0,347	0,343	0,330	0,339	0,301	0,304	0,277	0,300	0,291	0,283
	static	1,000	0,719	0,561	0,270	0,253	0,253	0,326	0,268	0,551	0,338	0,328	0,239	0,329	0,331
	all	1,000	0,672	0,511	0,318	0,339	0,332	0,342	0,326	0,337	0,320	0,296	0,298	0,328	0,292

Table B4. Summarization benchmark results for method using motion histogram comparison to detect key frames. Table presents results achieved using different histogram comparison functions with mean average histogram used as comparison histogram. Using 24 fps in intermediate videos.

						mean average									
						max deviance					min deviance				
	video	Upper bound	Mean human	SumMe	Random	bhat	chi	corr	int	l2	bhat	chi	corr	int	l2
Egocentric	Base jumping	0,398	0,257	0,121	0,144	0,086	0,066	0,139	0,062	0,078	0,13	0,134	0,17	0,132	0,206
	Bike Polo	0,503	0,322	0,356	0,134	0,082	0,075	0,079	0,083	0,133	0,18	0,184	0,117	0,184	0,153
	Scuba	0,387	0,217	0,184	0,138	0,196	0,179	0,137	0,203	0,154	0,189	0,17	0,219	0,174	0,186
	Valparaiso_Downhill	0,427	0,272	0,242	0,142	0,202	0,202	0,202	0,202	0,202	0,184	0,184	0,212	0,184	0,082
Moving	Bearpark_climbing	0,33	0,208	0,118	0,147	0,244	0,244	0,045	0,244	0,12	0,106	0,106	0,229	0,106	0,164
	Bus_in_Rock_Tunnel	0,359	0,198	0,135	0,135	0,09	0,087	0,121	0,097	0,129	0,097	0,097	0,13	0,097	0,116
	Car_railcrossing	0,515	0,357	0,362	0,14	0,144	0,097	0,07	0,158	0,119	0,096	0,1	0,19	0,099	0,051
	Cockpit_Landing	0,443	0,279	0,172	0,136	0,081	0,082	0,068	0,066	0,147	0,161	0,161	0,104	0,195	0,264
	Cooking	0,528	0,379	0,321	0,145	0,247	0,268	0,272	0,265	0,046	0,151	0,151	0,263	0,294	0,003
	Eiffel Tower	0,467	0,312	0,295	0,13	0,133	0,138	0,153	0,133	0,11	0,185	0,17	0,153	0,151	0,207
	Excavators river crossing	0,411	0,303	0,189	0,144	0,114	0,165	0,145	0,148	0,162	0,15	0,147	0,132	0,153	0,15
	Jumps	0,611	0,483	0,427	0,149	0,109	0,087	0,168	0,087	0,158	0,249	0,249	0,18	0,157	0,179
	Kids_playing_in_leaves	0,394	0,289	0,089	0,139	0,088	0,096	0,051	0,095	0,068	0,125	0,125	0,053	0,125	0,141
	Playing_on_water_slide	0,34	0,195	0,2	0,134	0,115	0,107	0,168	0,129	0,096	0,183	0,17	0,149	0,164	0,091
	Saving dolphines	0,313	0,188	0,145	0,144	0,29	0,111	0,29	0,29	0,108	0,28	0,28	0,168	0,28	0,122
	St Maarten Landing	0,624	0,496	0,313	0,143	0,21	0,215	0,036	0,214	0,191	0,062	0,062	0,185	0,033	0,166
	Statue of Liberty	0,332	0,184	0,192	0,122	0,108	0,141	0,134	0,115	0,185	0,078	0,081	0,113	0,083	0,112
	Uncut_Evening_Flight	0,506	0,35	0,271	0,131	0,091	0,095	0,259	0,113	0,07	0,222	0,276	0,122	0,194	0,174
	paluma_jump	0,662	0,509	0,181	0,139	0,006	0,047	0,011	0,047	0	0,104	0,104	0,104	0,104	0,112
	playing_ball	0,403	0,271	0,174	0,145	0,163	0,163	0,113	0,157	0,226	0,087	0,087	0,151	0,089	0,086
	Notre_Dame	0,36	0,231	0,235	0,137	0,128	0,104	0,118	0,115	0,136	0,143	0,136	0,162	0,155	0,139
static	Air_Force_One	0,49	0,332	0,318	0,144	0,103	0,025	0,065	0,103	0,324	0,29	0,29	0,289	0,058	0,091
	Fire Domino	0,514	0,394	0,13	0,145	0,041	0,041	0,071	0,041	0,506	0,148	0,148	0,038	0,107	0,306
	car_over_camera	0,49	0,346	0,372	0,134	0,069	0,068	0,07	0,068	0,104	0,201	0,201	0,25	0,155	0,243
	Paintball	0,55	0,399	0,32	0,127	0,218	0,088	0,091	0,215	0,255	0,057	0,055	0,068	0,055	0,191
						normalized to upper bound									
						max deviance					min deviance				
	video	Upper bound	Mean human	SumMe	Random	bhat	chi	corr	int	l2	bhat	chi	corr	int	l2
Egocentric	Base jumping	1,000	0,646	0,304	0,362	0,216	0,166	0,349	0,156	0,196	0,327	0,337	0,427	0,332	0,518
	Bike Polo	1,000	0,640	0,708	0,266	0,163	0,149	0,157	0,165	0,264	0,358	0,366	0,233	0,366	0,304
	Scuba	1,000	0,561	0,475	0,357	0,506	0,463	0,354	0,525	0,398	0,488	0,439	0,566	0,450	0,481
	Valparaiso_Downhill	1,000	0,637	0,567	0,333	0,473	0,473	0,473	0,473	0,473	0,431	0,431	0,496	0,431	0,192
Moving	Bearpark_climbing	1,000	0,630	0,358	0,445	0,739	0,739	0,136	0,739	0,364	0,321	0,321	0,694	0,321	0,497
	Bus_in_Rock_Tunnel	1,000	0,552	0,376	0,376	0,251	0,242	0,337	0,270	0,359	0,270	0,270	0,362	0,270	0,323
	Car_railcrossing	1,000	0,693	0,703	0,272	0,280	0,188	0,136	0,307	0,231	0,186	0,194	0,369	0,192	0,099
	Cockpit_Landing	1,000	0,630	0,388	0,307	0,183	0,185	0,153	0,149	0,332	0,363	0,363	0,235	0,440	0,596
	Cooking	1,000	0,718	0,608	0,275	0,468	0,508	0,515	0,502	0,087	0,286	0,286	0,498	0,557	0,006
	Eiffel Tower	1,000	0,668	0,632	0,278	0,285	0,296	0,328	0,285	0,236	0,396	0,364	0,328	0,323	0,443
	Excavators river crossing	1,000	0,737	0,460	0,350	0,277	0,401	0,353	0,360	0,394	0,365	0,358	0,321	0,372	0,365
	Jumps	1,000	0,791	0,699	0,244	0,178	0,142	0,275	0,142	0,259	0,408	0,408	0,295	0,257	0,293
	Kids_playing_in_leaves	1,000	0,734	0,226	0,353	0,223	0,244	0,129	0,241	0,173	0,317	0,317	0,135	0,317	0,358
	Playing_on_water_slide	1,000	0,574	0,588	0,394	0,338	0,315	0,494	0,379	0,282	0,538	0,500	0,438	0,482	0,268
	Saving dolphines	1,000	0,601	0,463	0,460	0,927	0,355	0,927	0,927	0,345	0,895	0,895	0,537	0,895	0,390
	St Maarten Landing	1,000	0,795	0,502	0,229	0,337	0,345	0,058	0,343	0,306	0,099	0,099	0,296	0,053	0,266
	Statue of Liberty	1,000	0,554	0,578	0,367	0,325	0,425	0,404	0,346	0,557	0,235	0,244	0,340	0,250	0,337
	Uncut_Evening_Flight	1,000	0,692	0,536	0,259	0,180	0,188	0,512	0,223	0,138	0,439	0,545	0,241	0,383	0,344
	paluma_jump	1,000	0,769	0,273	0,210	0,009	0,071	0,017	0,071	0,000	0,157	0,157	0,157	0,157	0,169
	playing_ball	1,000	0,672	0,432	0,360	0,404	0,404	0,280	0,390	0,561	0,216	0,216	0,375	0,221	0,213
	Notre_Dame	1,000	0,642	0,653	0,381	0,356	0,289	0,328	0,319	0,378	0,397	0,378	0,450	0,431	0,386
static	Air_Force_One	1,000	0,678	0,649	0,294	0,210	0,051	0,133	0,210	0,661	0,592	0,592	0,590	0,118	0,186
	Fire Domino	1,000	0,767	0,253	0,282	0,080	0,080	0,138	0,080	0,984	0,288	0,288	0,074	0,208	0,595
	car_over_camera	1,000	0,706	0,759	0,273	0,141	0,139	0,143	0,139	0,212	0,410	0,410	0,510	0,316	0,496
	Paintball	1,000	0,725	0,582	0,231	0,396	0,160	0,165	0,391	0,464	0,104	0,100	0,124	0,100	0,347
Mean	egosentric	1,000	0,621	0,513	0,329	0,340	0,313	0,333	0,330	0,333	0,401	0,393	0,431	0,394	0,374
	moving	1,000	0,674	0,498	0,327	0,339	0,314	0,317	0,353	0,294	0,346	0,348	0,357	0,348	0,315
	static	1,000	0,719	0,561	0,270	0,207	0,107	0,145	0,205	0,580	0,348	0,347	0,324	0,186	0,406
	all	1,000	0,672	0,511	0,318	0,318	0,281	0,292	0,325	0,346	0,355	0,355	0,364	0,330	0,339

Table B5. Summarization benchmark results for method using motion histogram comparison to detect key frames. Table presents results achieved using different histogram comparison functions with median average histogram used as comparison histogram. Using 24 fps in intermediate videos.

						median average									
						max deviance					min deviance				
	video	Upper bound	Mean human	SumMe	Random	bhat	chi	corr	int	l2	bhat	chi	corr	int	l2
Egocentric	Base jumping	0,398	0,257	0,121	0,144	0,164	0,125	0,15	0,152	0,088	0,216	0,22	0,099	0,227	0,071
	Bike Polo	0,503	0,322	0,356	0,134	0,083	0,048	0,086	0,15	0,134	0,118	0,112	0,186	0,249	0,22
	Scuba	0,387	0,217	0,184	0,138	0,191	0,211	0,124	0,158	0,128	0,162	0,175	0,125	0,126	0,148
	Valparaiso_Downhill	0,427	0,272	0,242	0,142	0,202	0,202	0,207	0,202	0,202	0,184	0,184	0,067	0,184	0,184
Moving	Bearpark_climbing	0,33	0,208	0,118	0,147	0,244	0,247	0,04	0,247	0,12	0,195	0,195	0,222	0,199	0,223
	Bus_in_Rock_Tunnel	0,359	0,198	0,135	0,135	0,107	0,092	0,087	0,108	0,126	0,102	0,101	0,124	0,101	0,112
	Car_railcrossing	0,515	0,357	0,362	0,14	0,115	0,069	0,113	0,065	0,112	0,151	0,148	0,189	0,044	0,124
	Cockpit_Landing	0,443	0,279	0,172	0,136	0,127	0,122	0,075	0,093	0,15	0,204	0,21	0,208	0,203	0,197
	Cooking	0,528	0,379	0,321	0,145	0,177	0,172	0,273	0,23	0,046	0,185	0,185	0,189	0,185	0,091
	Eiffel Tower	0,467	0,312	0,295	0,13	0,152	0,135	0,127	0,14	0,112	0,081	0,085	0,075	0,074	0,15
	Excavators river crossing	0,411	0,303	0,189	0,144	0,127	0,134	0,145	0,126	0,163	0,136	0,129	0,115	0,127	0,141
	Jumps	0,611	0,483	0,427	0,149	0,144	0,14	0,108	0,14	0,183	0,262	0,236	0,252	0,236	0,169
	Kids_playing_in_leaves	0,394	0,289	0,089	0,139	0,089	0,09	0,246	0,09	0,068	0,235	0,235	0,34	0,32	0,314
	Playing_on_water_slide	0,34	0,195	0,2	0,134	0,074	0,097	0,145	0,069	0,065	0,22	0,22	0,174	0,219	0,186
	Saving dolphines	0,313	0,188	0,145	0,144	0,29	0,111	0,29	0,29	0,108	0,23	0,12	0,28	0,12	0,23
	St Maarten Landing	0,624	0,496	0,313	0,143	0,211	0,21	0,15	0,211	0,191	0,078	0,057	0,092	0,059	0,064
	Statue of Liberty	0,332	0,184	0,192	0,122	0,109	0,115	0,116	0,104	0,18	0,17	0,167	0,13	0,12	0,089
	Uncut_Evening_Flight	0,506	0,35	0,271	0,131	0,08	0,091	0,121	0,096	0,065	0,083	0,089	0,188	0,195	0,11
	paluma_jump	0,662	0,509	0,181	0,139	0,047	0,047	0,001	0,047	0	0,017	0,017	0,017	0,017	0,017
	playing_ball	0,403	0,271	0,174	0,145	0,159	0,162	0,178	0,161	0,232	0,069	0,069	0,167	0,065	0,143
	Notre_Dame	0,36	0,231	0,235	0,137	0,123	0,135	0,139	0,11	0,137	0,135	0,135	0,146	0,129	0,115
static	Air_Force_One	0,49	0,332	0,318	0,144	0,103	0,025	0,065	0,103	0,324	0,11	0,11	0,048	0,11	0,065
	Fire Domino	0,514	0,394	0,13	0,145	0,041	0,041	0,252	0,044	0,506	0,027	0,027	0,028	0,087	0,101
	car_over_camera	0,49	0,346	0,372	0,134	0,068	0,069	0,111	0,068	0,104	0,066	0,199	0,068	0,063	0,05
	Paintball	0,55	0,399	0,32	0,127	0,149	0,149	0,16	0,149	0,255	0,153	0,153	0,153	0,153	0,152
		normalized to upper bound													
		max deviance									min deviance				
	video	Upper bound	Mean human	SumMe	Random	bhat	chi	corr	int	l2	bhat	chi	corr	int	l2
Egocentric	Base jumping	1,000	0,646	0,304	0,362	0,412	0,314	0,377	0,382	0,221	0,543	0,553	0,249	0,570	0,178
	Bike Polo	1,000	0,640	0,708	0,266	0,165	0,095	0,171	0,298	0,266	0,235	0,223	0,370	0,495	0,437
	Scuba	1,000	0,561	0,475	0,357	0,494	0,545	0,320	0,408	0,331	0,419	0,452	0,323	0,326	0,382
	Valparaiso_Downhill	1,000	0,637	0,567	0,333	0,473	0,473	0,485	0,473	0,473	0,431	0,431	0,157	0,431	0,431
Moving	Bearpark_climbing	1,000	0,630	0,358	0,445	0,739	0,748	0,121	0,748	0,364	0,591	0,591	0,673	0,603	0,676
	Bus_in_Rock_Tunnel	1,000	0,552	0,376	0,376	0,298	0,256	0,242	0,301	0,351	0,284	0,281	0,345	0,281	0,312
	Car_railcrossing	1,000	0,693	0,703	0,272	0,223	0,134	0,219	0,126	0,217	0,293	0,287	0,367	0,085	0,241
	Cockpit_Landing	1,000	0,630	0,388	0,307	0,287	0,275	0,169	0,210	0,339	0,460	0,474	0,470	0,458	0,445
	Cooking	1,000	0,718	0,608	0,275	0,335	0,326	0,517	0,436	0,087	0,350	0,350	0,358	0,350	0,172
	Eiffel Tower	1,000	0,668	0,632	0,278	0,325	0,289	0,272	0,300	0,240	0,173	0,182	0,161	0,158	0,321
	Excavators river crossing	1,000	0,737	0,460	0,350	0,309	0,326	0,353	0,307	0,397	0,331	0,314	0,280	0,309	0,343
	Jumps	1,000	0,791	0,699	0,244	0,236	0,229	0,177	0,229	0,300	0,429	0,386	0,412	0,386	0,277
	Kids_playing_in_leaves	1,000	0,734	0,226	0,353	0,226	0,228	0,624	0,228	0,173	0,596	0,596	0,863	0,812	0,797
	Playing_on_water_slide	1,000	0,574	0,588	0,394	0,218	0,285	0,426	0,203	0,191	0,647	0,647	0,512	0,644	0,547
	Saving dolphines	1,000	0,601	0,463	0,460	0,927	0,355	0,927	0,927	0,345	0,735	0,383	0,895	0,383	0,735
	St Maarten Landing	1,000	0,795	0,502	0,229	0,338	0,337	0,240	0,338	0,306	0,125	0,091	0,147	0,095	0,103
	Statue of Liberty	1,000	0,554	0,578	0,367	0,328	0,346	0,349	0,313	0,542	0,512	0,503	0,392	0,361	0,268
	Uncut_Evening_Flight	1,000	0,692	0,536	0,259	0,158	0,180	0,239	0,190	0,128	0,164	0,176	0,372	0,385	0,217
	paluma_jump	1,000	0,769	0,273	0,210	0,071	0,071	0,002	0,071	0,000	0,026	0,026	0,026	0,026	0,026
	playing_ball	1,000	0,672	0,432	0,360	0,395	0,402	0,442	0,400	0,576	0,171	0,171	0,414	0,161	0,355
	Notre_Dame	1,000	0,642	0,653	0,381	0,342	0,375	0,386	0,306	0,381	0,375	0,375	0,406	0,358	0,319
static	Air_Force_One	1,000	0,678	0,649	0,294	0,210	0,051	0,133	0,210	0,661	0,224	0,224	0,098	0,224	0,133
	Fire Domino	1,000	0,767	0,253	0,282	0,080	0,080	0,490	0,086	0,984	0,053	0,053	0,054	0,169	0,196
	car_over_camera	1,000	0,706	0,759	0,273	0,139	0,141	0,227	0,139	0,212	0,135	0,406	0,139	0,129	0,102
	Paintball	1,000	0,725	0,582	0,231	0,271	0,271	0,291	0,271	0,464	0,278	0,278	0,278	0,278	0,276
Mean	egosentric	1,000	0,621	0,513	0,329	0,386	0,357	0,338	0,390	0,323	0,407	0,415	0,275	0,455	0,357
	moving	1,000	0,674	0,498	0,327	0,339	0,304	0,336	0,331	0,290	0,368	0,343	0,417	0,345	0,362
	static	1,000	0,719	0,561	0,270	0,175	0,136	0,285	0,176	0,580	0,172	0,240	0,142	0,200	0,177
	all	1,000	0,672	0,511	0,318	0,320	0,285	0,328	0,316	0,342	0,343	0,338	0,350	0,339	0,332

Table B6. Summarization benchmark results for method using motion histogram comparison to detect key frames. Table presents results achieved using different histogram comparison functions with mean average histogram used as comparison histogram. Using 15 fps in intermediate videos.

						mean average										
						max deviance					min deviance					
	video	Upper bound	Mean human	SumMe	Random	bhat	chi	corr	int	l2	bhat	chi	corr	int	l2	
Egocentric	Base jumping	0,398	0,257	0,121	0,144	0,078	0,077	0,205	0,078	0,079	0,104	0,104	0,247	0,087	0,133	
	Bike Polo	0,503	0,322	0,356	0,134	0,059	0,12	0,099	0,06	0,104	0,195	0,195	0,115	0,195	0,163	
	Scuba	0,387	0,217	0,184	0,138	0,203	0,197	0,119	0,205	0,193	0,151	0,163	0,17	0,173	0,134	
	Valparaiso_Downhill	0,427	0,272	0,242	0,142	0,226	0,192	0,113	0,221	0,197	0,203	0,237	0,177	0,237	0,102	
Moving	Bearpark_climbing	0,33	0,208	0,118	0,147	0,204	0,141	0,131	0,215	0,203	0,134	0,134	0,099	0,13	0,123	
	Bus_in_Rock_Tunnel	0,359	0,198	0,135	0,135	0,208	0,191	0,143	0,176	0,145	0,15	0,184	0,118	0,183	0,179	
	Car_railcrossing	0,515	0,357	0,362	0,14	0,148	0,144	0,077	0,142	0,106	0,217	0,213	0,058	0,06	0,059	
	Cockpit_Landing	0,443	0,279	0,172	0,136	0,129	0,129	0,081	0,124	0,149	0,223	0,232	0,146	0,22	0,185	
	Cooking	0,528	0,379	0,321	0,145	0,313	0,318	0,189	0,313	0,034	0,073	0,073	0,131	0,042	0,013	
	Eiffel Tower	0,467	0,312	0,295	0,13	0,103	0,152	0,114	0,097	0,2	0,107	0,133	0,149	0,142	0,18	
	Excavators river crossing	0,411	0,303	0,189	0,144	0,167	0,162	0,166	0,16	0,148	0,15	0,145	0,131	0,149	0,119	
	Jumps	0,611	0,483	0,427	0,149	0,06	0,06	0,119	0,06	0,231	0,161	0,161	0,124	0,161	0,154	
	Kids_playing_in_leaves	0,394	0,289	0,089	0,139	0,054	0,054	0,424	0,054	0,076	0,076	0,076	0,072	0,076	0,076	
	Playing_on_water_slide	0,34	0,195	0,2	0,134	0,121	0,068	0,218	0,075	0,057	0,224	0,227	0,129	0,203	0,149	
	Saving dolphines	0,313	0,188	0,145	0,144	0,196	0,197	0,238	0,196	0,068	0,108	0,108	0,127	0,125	0,148	
	St Maarten Landing	0,624	0,496	0,313	0,143	0,287	0,284	0,18	0,284	0,104	0,112	0,058	0,091	0,083	0,037	
	Statue of Liberty	0,332	0,184	0,192	0,122	0,123	0,13	0,12	0,12	0,132	0,097	0,076	0,116	0,097	0,114	
	Uncut_Evening_Flight	0,506	0,35	0,271	0,131	0,07	0,076	0,201	0,102	0,096	0,093	0,077	0,08	0,169	0,215	
	paluma_jump	0,662	0,509	0,181	0,139	0,017	0,017	0,028	0,017	0,398	0,105	0,105	0,105	0,105	0,105	
	playing_ball	0,403	0,271	0,174	0,145	0,133	0,066	0,11	0,133	0,314	0,054	0,054	0,056	0,054	0,054	
	Notre_Dame	0,36	0,231	0,235	0,137	0,134	0,13	0,12	0,132	0,157	0,118	0,131	0,153	0,142	0,132	
static	Air_Force_One	0,49	0,332	0,318	0,144	0,061	0,055	0,065	0,051	0,32	0,298	0,298	0,297	0,022	0,354	
	Fire Domino	0,514	0,394	0,13	0,145	0,178	0,426	0,174	0,143	0,28	0,257	0,257	0,031	0,368	0,267	
	car_over_camera	0,49	0,346	0,372	0,134	0,073	0,073	0,058	0,073	0,115	0,231	0,245	0,196	0,245	0,227	
	Paintball	0,55	0,399	0,32	0,127	0,219	0,093	0,067	0,069	0,253	0,076	0,125	0,159	0,129	0,077	
						normalized to upper bound										
						max deviance					min deviance					
	video	Upper bound	Mean human	SumMe	Random	bhat	chi	corr	int	l2	bhat	chi	corr	int	l2	
Egocentric	Base jumping	1,000	0,646	0,304	0,362	0,196	0,193	0,515	0,196	0,198	0,261	0,261	0,621	0,219	0,334	
	Bike Polo	1,000	0,640	0,708	0,266	0,117	0,239	0,197	0,119	0,207	0,388	0,388	0,229	0,388	0,324	
	Scuba	1,000	0,561	0,475	0,357	0,525	0,509	0,307	0,530	0,499	0,390	0,421	0,439	0,447	0,346	
	Valparaiso_Downhill	1,000	0,637	0,567	0,333	0,529	0,450	0,265	0,518	0,461	0,475	0,555	0,415	0,555	0,239	
Moving	Bearpark_climbing	1,000	0,630	0,358	0,445	0,618	0,427	0,397	0,652	0,615	0,406	0,406	0,300	0,394	0,373	
	Bus_in_Rock_Tunnel	1,000	0,552	0,376	0,376	0,579	0,532	0,398	0,490	0,404	0,418	0,513	0,329	0,510	0,499	
	Car_railcrossing	1,000	0,693	0,703	0,272	0,287	0,280	0,150	0,276	0,206	0,421	0,414	0,113	0,117	0,115	
	Cockpit_Landing	1,000	0,630	0,388	0,307	0,291	0,291	0,183	0,280	0,336	0,503	0,524	0,330	0,497	0,418	
	Cooking	1,000	0,718	0,608	0,275	0,593	0,602	0,358	0,593	0,064	0,138	0,138	0,248	0,080	0,025	
	Eiffel Tower	1,000	0,668	0,632	0,278	0,221	0,325	0,244	0,208	0,428	0,229	0,285	0,319	0,304	0,385	
	Excavators river crossing	1,000	0,737	0,460	0,350	0,406	0,394	0,404	0,389	0,360	0,365	0,353	0,319	0,363	0,290	
	Jumps	1,000	0,791	0,699	0,244	0,098	0,098	0,195	0,098	0,378	0,264	0,264	0,203	0,264	0,252	
	Kids_playing_in_leaves	1,000	0,734	0,226	0,353	0,137	0,137	1,076	0,137	0,193	0,193	0,193	0,183	0,193	0,193	
	Playing_on_water_slide	1,000	0,574	0,588	0,394	0,356	0,200	0,641	0,221	0,168	0,659	0,668	0,379	0,597	0,438	
	Saving dolphines	1,000	0,601	0,463	0,460	0,626	0,629	0,760	0,626	0,217	0,345	0,345	0,406	0,399	0,473	
	St Maarten Landing	1,000	0,795	0,502	0,229	0,460	0,455	0,288	0,455	0,167	0,179	0,093	0,146	0,133	0,059	
	Statue of Liberty	1,000	0,554	0,578	0,367	0,370	0,392	0,361	0,361	0,398	0,292	0,229	0,349	0,292	0,343	
	Uncut_Evening_Flight	1,000	0,692	0,536	0,259	0,138	0,150	0,397	0,202	0,190	0,184	0,152	0,158	0,334	0,425	
	paluma_jump	1,000	0,769	0,273	0,210	0,026	0,026	0,042	0,026	0,601	0,159	0,159	0,159	0,159	0,159	
	playing_ball	1,000	0,672	0,432	0,360	0,330	0,164	0,273	0,330	0,779	0,134	0,134	0,139	0,134	0,134	
	Notre_Dame	1,000	0,642	0,653	0,381	0,372	0,361	0,333	0,367	0,436	0,328	0,364	0,425	0,394	0,367	
static	Air_Force_One	1,000	0,678	0,649	0,294	0,124	0,112	0,133	0,104	0,653	0,608	0,608	0,606	0,045	0,722	
	Fire Domino	1,000	0,767	0,253	0,282	0,346	0,829	0,339	0,278	0,545	0,500	0,500	0,060	0,716	0,519	
	car_over_camera	1,000	0,706	0,759	0,273	0,149	0,149	0,118	0,149	0,235	0,471	0,500	0,400	0,500	0,463	
	Paintball	1,000	0,725	0,582	0,231	0,398	0,169	0,122	0,125	0,460	0,138	0,227	0,289	0,235	0,140	
Mean	egosentric	1,000	0,621	0,513	0,329	0,342	0,348	0,321	0,341	0,341	0,379	0,406	0,426	0,402	0,311	
	moving	1,000	0,674	0,498	0,327	0,348	0,321	0,382	0,336	0,349	0,307	0,308	0,265	0,304	0,291	
	static	1,000	0,719	0,561	0,270	0,254	0,315	0,178	0,164	0,473	0,429	0,459	0,339	0,374	0,461	
	all	1,000	0,672	0,511	0,318	0,332	0,325	0,340	0,309	0,368	0,338	0,348	0,302	0,331	0,321	

Table B7. Summarization benchmark results for method using motion histogram comparison to detect key frames. Table presents results achieved using different histogram comparison functions with median average histogram used as comparison histogram. Using 15 fps in intermediate videos.

						median average									
						max deviance					min deviance				
	video	Upper bound	Mean human	SumMe	Random	bhat	chi	corr	int	l2	bhat	chi	corr	int	l2
Egocentric	Base jumping	0,398	0,257	0,121	0,144	0,059	0,077	0,221	0,052	0,057	0,087	0,201	0,225	0,241	0,19
	Bike Polo	0,503	0,322	0,356	0,134	0,13	0,121	0,102	0,063	0,104	0,171	0,171	0,122	0,175	0,092
	Scuba	0,387	0,217	0,184	0,138	0,196	0,189	0,178	0,198	0,178	0,187	0,18	0,178	0,187	0,145
	Valparaiso_Downhill	0,427	0,272	0,242	0,142	0,139	0,192	0,108	0,184	0,197	0,086	0,086	0,183	0,149	0,115
Moving	Bearpark_climbing	0,33	0,208	0,118	0,147	0,147	0,137	0,162	0,222	0,2	0,155	0,16	0,113	0,126	0,143
	Bus_in_Rock_Tunnel	0,359	0,198	0,135	0,135	0,211	0,137	0,107	0,146	0,145	0,143	0,156	0,125	0,143	0,098
	Car_railcrossing	0,515	0,357	0,362	0,14	0,143	0,118	0,052	0,147	0,106	0,206	0,206	0,061	0,212	0,07
	Cockpit_Landing	0,443	0,279	0,172	0,136	0,118	0,093	0,084	0,134	0,161	0,154	0,147	0,191	0,15	0,103
	Cooking	0,528	0,379	0,321	0,145	0,225	0,023	0,105	0,226	0,034	0,255	0,1	0,136	0,211	0,094
	Eiffel Tower	0,467	0,312	0,295	0,13	0,155	0,139	0,089	0,116	0,197	0,137	0,067	0,07	0,144	0,091
	Excavators river crossing	0,411	0,303	0,189	0,144	0,134	0,134	0,166	0,135	0,144	0,148	0,141	0,132	0,141	0,167
	Jumps	0,611	0,483	0,427	0,149	0,102	0,133	0,058	0,13	0,13	0,243	0,243	0,187	0,243	0,262
	Kids_playing_in_leaves	0,394	0,289	0,089	0,139	0,049	0,049	0,424	0,049	0,082	0,076	0,11	0,076	0,11	0,424
	Playing_on_water_slide	0,34	0,195	0,2	0,134	0,082	0,138	0,13	0,082	0,057	0,169	0,169	0,162	0,172	0,14
	Saving dolphines	0,313	0,188	0,145	0,144	0,196	0,197	0,179	0,196	0,18	0,149	0,153	0,15	0,153	0,194
	St Maarten Landing	0,624	0,496	0,313	0,143	0,241	0,24	0,177	0,248	0,104	0,03	0,052	0,042	0,124	0,02
	Statue of Liberty	0,332	0,184	0,192	0,122	0,12	0,12	0,125	0,119	0,131	0,131	0,129	0,127	0,132	0,127
	Uncut_Evening_Flight	0,506	0,35	0,271	0,131	0,087	0,072	0,131	0,089	0,131	0,194	0,211	0,17	0,211	0,121
	paluma_jump	0,662	0,509	0,181	0,139	0	0	0,287	0	0,398	0,429	0,429	0,429	0,429	0,429
	playing_ball	0,403	0,271	0,174	0,145	0,133	0,066	0,168	0,133	0,314	0,054	0,054	0,049	0,196	0,196
	Notre_Dame	0,36	0,231	0,235	0,137	0,153	0,138	0,126	0,138	0,161	0,11	0,109	0,137	0,111	0,129
static	Air_Force_One	0,49	0,332	0,318	0,144	0,061	0,055	0,065	0,051	0,32	0,024	0,024	0,11	0,065	0,359
	Fire Domino	0,514	0,394	0,13	0,145	0,054	0,259	0,124	0,054	0,29	0,179	0,179	0,091	0,176	0,146
	car_over_camera	0,49	0,346	0,372	0,134	0,073	0,073	0,052	0,073	0,115	0,069	0,066	0,159	0,154	0,075
	Paintball	0,55	0,399	0,32	0,127	0,155	0,155	0,162	0,154	0,253	0,209	0,206	0,207	0,208	0,059
		normalized to upper bound													
						max deviance					min deviance				
	video	Upper bound	Mean human	SumMe	Random	bhat	chi	corr	int	l2	bhat	chi	corr	int	l2
Egocentric	Base jumping	1,000	0,646	0,304	0,362	0,148	0,193	0,555	0,131	0,143	0,219	0,505	0,565	0,606	0,477
	Bike Polo	1,000	0,640	0,708	0,266	0,258	0,241	0,203	0,125	0,207	0,340	0,340	0,243	0,348	0,183
	Scuba	1,000	0,561	0,475	0,357	0,506	0,488	0,460	0,512	0,460	0,483	0,465	0,460	0,483	0,375
	Valparaiso_Downhill	1,000	0,637	0,567	0,333	0,326	0,450	0,253	0,431	0,461	0,201	0,201	0,429	0,349	0,269
Moving	Bearpark_climbing	1,000	0,630	0,358	0,445	0,445	0,415	0,491	0,673	0,606	0,470	0,485	0,342	0,382	0,433
	Bus_in_Rock_Tunnel	1,000	0,552	0,376	0,376	0,588	0,382	0,298	0,407	0,404	0,398	0,435	0,348	0,398	0,273
	Car_railcrossing	1,000	0,693	0,703	0,272	0,278	0,229	0,101	0,285	0,206	0,400	0,400	0,118	0,412	0,136
	Cockpit_Landing	1,000	0,630	0,388	0,307	0,266	0,210	0,190	0,302	0,363	0,348	0,332	0,431	0,339	0,233
	Cooking	1,000	0,718	0,608	0,275	0,426	0,044	0,199	0,428	0,064	0,483	0,189	0,258	0,400	0,178
	Eiffel Tower	1,000	0,668	0,632	0,278	0,332	0,298	0,191	0,248	0,422	0,293	0,143	0,150	0,308	0,195
	Excavators river crossing	1,000	0,737	0,460	0,350	0,326	0,326	0,404	0,328	0,350	0,360	0,343	0,321	0,343	0,406
	Jumps	1,000	0,791	0,699	0,244	0,167	0,218	0,095	0,213	0,213	0,398	0,398	0,306	0,398	0,429
	Kids_playing_in_leaves	1,000	0,734	0,226	0,353	0,124	0,124	1,076	0,124	0,208	0,193	0,279	0,193	0,279	1,076
	Playing_on_water_slide	1,000	0,574	0,588	0,394	0,241	0,406	0,382	0,241	0,168	0,497	0,497	0,476	0,506	0,412
	Saving dolphines	1,000	0,601	0,463	0,460	0,626	0,629	0,572	0,626	0,575	0,476	0,489	0,479	0,489	0,620
	St Maarten Landing	1,000	0,795	0,502	0,229	0,386	0,385	0,284	0,397	0,167	0,048	0,083	0,067	0,199	0,032
	Statue of Liberty	1,000	0,554	0,578	0,367	0,361	0,361	0,377	0,358	0,395	0,395	0,389	0,383	0,398	0,383
	Uncut_Evening_Flight	1,000	0,692	0,536	0,259	0,172	0,142	0,259	0,176	0,259	0,383	0,417	0,336	0,417	0,239
	paluma_jump	1,000	0,769	0,273	0,210	0,000	0,000	0,434	0,000	0,601	0,648	0,648	0,648	0,648	0,648
	playing_ball	1,000	0,672	0,432	0,360	0,330	0,164	0,417	0,330	0,779	0,134	0,134	0,122	0,486	0,486
	Notre_Dame	1,000	0,642	0,653	0,381	0,425	0,383	0,350	0,383	0,447	0,306	0,303	0,381	0,308	0,358
static	Air_Force_One	1,000	0,678	0,649	0,294	0,124	0,112	0,133	0,104	0,653	0,049	0,049	0,224	0,133	0,733
	Fire Domino	1,000	0,767	0,253	0,282	0,105	0,504	0,241	0,105	0,564	0,348	0,348	0,177	0,342	0,284
	car_over_camera	1,000	0,706	0,759	0,273	0,149	0,149	0,106	0,149	0,235	0,141	0,135	0,324	0,314	0,153
	Paintball	1,000	0,725	0,582	0,231	0,282	0,282	0,295	0,280	0,460	0,380	0,375	0,376	0,378	0,107
Mean	egosentric	1,000	0,621	0,513	0,329	0,310	0,343	0,368	0,300	0,318	0,311	0,378	0,424	0,446	0,326
	moving	1,000	0,674	0,498	0,327	0,323	0,277	0,360	0,325	0,366	0,366	0,351	0,315	0,395	0,385
	static	1,000	0,719	0,561	0,270	0,165	0,262	0,194	0,160	0,478	0,230	0,227	0,276	0,292	0,319
	all	1,000	0,672	0,511	0,318	0,296	0,285	0,335	0,294	0,376	0,336	0,335	0,326	0,386	0,365

APPENDIX C. Summarization benchmark results (human selected key frames)

Table C1. Summarization benchmark results for method using human selected key frames. Table presents results achieved using annotations from test subjects a, b, c, d and e. Subject b did not annotate all the videos in the test set.

	video	Upper bound	Mean human	SumMe	Random	a	b	c	d	e	Mean	Mean time
Egocentric	Base jumping	0,398	0,257	0,121	0,144	0,201	0,143	0,18	0,144	0,154	0,164	01:27
	Bike Polo	0,503	0,322	0,356	0,134	0,143	0,085	0,104	0,299	0,069	0,140	01:10
	Scuba	0,387	0,217	0,184	0,138	0,115	0,116	0,155	0,073	0,156	0,123	00:47
	Valparaiso_Downhill	0,427	0,272	0,242	0,142	0,152		0,289	0,182	0,218	0,210	00:18
Moving	Bearpark_climbing	0,33	0,208	0,118	0,147	0,17	0,208	0,179	0,205	0,18	0,188	00:28
	Bus_in_Rock_Tunnel	0,359	0,198	0,135	0,135	0,161	0,22	0,135	0,171	0,122	0,162	00:54
	Car_railcrossing	0,515	0,357	0,362	0,14	0,307	0,126	0,082	0,119	0,238	0,174	00:55
	Cockpit_Landing	0,443	0,279	0,172	0,136	0,159	0,251	0,191	0,345	0,145	0,218	01:12
	Cooking	0,528	0,379	0,321	0,145	0,372	0,245	0,104	0,088	0,11	0,184	00:26
	Eiffel Tower	0,467	0,312	0,295	0,13	0,149	0,283	0,122	0,17	0,209	0,187	00:55
	Excavators river crossing	0,411	0,303	0,189	0,144	0,258	0,385	0,14	0,194	0,145	0,224	01:46
	Jumps	0,611	0,483	0,427	0,149	0,483	0,141	0,194	0,136	0,118	0,214	00:27
	Kids_playing_in_leaves	0,394	0,289	0,089	0,139	0,259	0,247	0,345	0,146	0,24	0,247	00:18
	Playing_on_water_slide	0,34	0,195	0,2	0,134	0,159	0,173	0,175	0,169	0,15	0,165	00:37
	Saving dolphines	0,313	0,188	0,145	0,144	0,156		0,156	0,107	0,166	0,146	00:16
	St Maarten Landing	0,624	0,496	0,313	0,143	0,264	0,406	0,126	0,148	0,134	0,216	00:20
	Statue of Liberty	0,332	0,184	0,192	0,122	0,155	0,05	0,137	0,147	0,151	0,128	00:56
	Uncut_Evening_Flight	0,506	0,35	0,271	0,131	0,126		0,115	0,126	0,095	0,116	00:34
	paluma_jump	0,662	0,509	0,181	0,139	0,091	0	0,153	0,079	0,079	0,080	00:09
	playing_ball	0,403	0,271	0,174	0,145	0,124	0,193	0,138	0,186	0,128	0,154	00:37
	Notre_Dame	0,36	0,231	0,235	0,137	0,192	0,171	0,157	0,15	0,106	0,155	01:13
static	Air_Force_One	0,49	0,332	0,318	0,144	0,37	0,37	0,37	0,302	0,314	0,345	00:19
	Fire Domino	0,514	0,394	0,13	0,145	0,286	0,353	0,046	0,035	0,158	0,176	00:22
	car_over_camera	0,49	0,346	0,372	0,134	0,404	0,408	0,152	0,38	0,262	0,321	00:34
	Paintball	0,55	0,399	0,32	0,127	0,252	0,228	0,122	0,324	0,227	0,231	00:35
		Normalized to upper bound										
	video	Upper bound	Mean human	SumMe	Random	a	b	c	d	e	Mean	Mean time
Egocentric	Base jumping	1,000	0,646	0,304	0,362	0,505	0,359	0,452	0,362	0,387	0,413	01:27
	Bike Polo	1,000	0,640	0,708	0,266	0,284	0,169	0,207	0,594	0,137	0,278	01:10
	Scuba	1,000	0,561	0,475	0,357	0,297	0,300	0,401	0,189	0,403	0,318	00:47
	Valparaiso_Downhill	1,000	0,637	0,567	0,333	0,356		0,677	0,426	0,511	0,492	00:18
Moving	Bearpark_climbing	1,000	0,630	0,358	0,445	0,515	0,630	0,542	0,621	0,545	0,571	00:28
	Bus_in_Rock_Tunnel	1,000	0,552	0,376	0,376	0,448	0,613	0,376	0,476	0,340	0,451	00:54
	Car_railcrossing	1,000	0,693	0,703	0,272	0,596	0,245	0,159	0,231	0,462	0,339	00:55
	Cockpit_Landing	1,000	0,630	0,388	0,307	0,359	0,567	0,431	0,779	0,327	0,493	01:12
	Cooking	1,000	0,718	0,608	0,275	0,705	0,464	0,197	0,167	0,208	0,348	00:26
	Eiffel Tower	1,000	0,668	0,632	0,278	0,319	0,606	0,261	0,364	0,448	0,400	00:55
	Excavators river crossing	1,000	0,737	0,460	0,350	0,628	0,937	0,341	0,472	0,353	0,546	01:46
	Jumps	1,000	0,791	0,699	0,244	0,791	0,231	0,318	0,223	0,193	0,351	00:27
	Kids_playing_in_leaves	1,000	0,734	0,226	0,353	0,657	0,627	0,876	0,371	0,609	0,628	00:18
	Playing_on_water_slide	1,000	0,574	0,588	0,394	0,468	0,509	0,515	0,497	0,441	0,486	00:37
	Saving dolphines	1,000	0,601	0,463	0,460	0,498		0,498	0,342	0,530	0,467	00:16
	St Maarten Landing	1,000	0,795	0,502	0,229	0,423	0,651	0,202	0,237	0,215	0,346	00:20
	Statue of Liberty	1,000	0,554	0,578	0,367	0,467	0,151	0,413	0,443	0,455	0,386	00:56
	Uncut_Evening_Flight	1,000	0,692	0,536	0,259	0,249		0,227	0,249	0,188	0,228	00:34
	paluma_jump	1,000	0,769	0,273	0,210	0,137	0,000	0,231	0,119	0,119	0,121	00:09
	playing_ball	1,000	0,672	0,432	0,360	0,308	0,479	0,342	0,462	0,318	0,382	00:37
	Notre_Dame	1,000	0,642	0,653	0,381	0,533	0,475	0,436	0,417	0,294	0,431	01:13
static	Air_Force_One	1,000	0,678	0,649	0,294	0,755	0,755	0,755	0,616	0,641	0,704	00:19
	Fire Domino	1,000	0,767	0,253	0,282	0,556	0,687	0,089	0,068	0,307	0,342	00:22
	car_over_camera	1,000	0,706	0,759	0,273	0,824	0,833	0,310	0,776	0,535	0,656	00:34
	Paintball	1,000	0,725	0,582	0,231	0,458	0,415	0,222	0,589	0,413	0,419	00:35
Mean	egosentric	1,000	0,621	0,513	0,329	0,361	0,276	0,434	0,393	0,359	0,365	03:42
	moving	1,000	0,674	0,498	0,327	0,477	0,479	0,374	0,381	0,356	0,413	12:04
	static	1,000	0,719	0,561	0,270	0,649	0,672	0,344	0,512	0,474	0,530	01:50
	all	1,000	0,672	0,511	0,318	0,486	0,486	0,379	0,404	0,375	0,426	17:36